Antti Vitikainen

Yrityksen SaaS-palvelujen toimintavarmuuden kartoitus ja kehitys

Metropolia Ammattikorkeakoulu
Insinööri (AMK)
Tietotekniikan koulutusohjelma
Insinöörityö
11.4.2012

Metropolia

Tiivistelmä

Tekijä Otsikko Sivumäärä Aika	Antti Vitikainen Yrityksen SaaS-palvelujen toimintavarmuuden kartoitus ja kehitys 55 sivua + 10 liitettä 11.4.2012
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikan koulutusohjelma
Suuntautumisvaihtoehto	Tietoverkot
Ohjaaja	Yliopettaja Janne Salonen

Tämän insinöörityön tarkoituksena oli kartoittaa TypingMaster Finland Oy:n SaaS-palveluiden toimintavarmuus ja parantaa sitä mahdollisuuksien mukaan. Tavoitteena oli, että palveluiden vikasietoisuutta saataisiin parannettua.

Työssä selvitettiin, mitkä tekijät ovat olennaisimpia SaaS-palvelujen toimintavarmuutta ajatellen. Jo alkuvaiheissaa kävi ilmi, miten suuressa roolissa data on SaaS-palveluissa. Tämän takia työssä pureuduttiin hyvin laajalti datan varmuuskopiointiin ja palautukseen.

Insinöörityön käytännön osuudessa kartoitettiin ensin yrityksen SaaS-palvelujen varmuuskopiointijärjestelmä, joka todettiin puutteelliseksi. Tämän jälkeen järjestelmä rakennettiin uusiksi saatavilla olevia resursseja hyväksi käyttäen. SaaS-palveluille rakennettiin Cold Standby ja Hot Standby -palautusjärjestelmien välimaastossa oleva järjestelmä MySQL-replikaation avulla. Periaatteena oli, että molemmilla SaaS-palvelimilla olisi samat konfiguraatiot, ohjelmistot ja data, mutta vain toinen olisi aktiivisena kerrallaan. Molemmissa silti on jatkuvasti aktiivisia palveluja, joten järjestelmästä tehtiin ristikkäinen.

Replikaation lisäksi rakennettiin varmuuskopiointijärjestelmä, joka lataa palveluiden päivittäiset varmuuskopiot talteen. Tämän jälkeen ne myös arkistoidaan asianmukaisesti. Sama järjestelmä implementoitiin myös yrityksen sisäisiin palveluihin, sillä muun muassa asiakasdata todettiin SaaS-palveluilla elintärkeäksi.

Yrityksen SaaS-palveluiden uhat kartoitettiin vielä tehtyjen muutosten jälkeen. Myös mahdolliseen palautumiseen varauduttiin niin, että palvelujen palauttaminen olisi mahdollisimman yksinkertainen toimenpide.

Tavoitteisiin päästiin, sillä yrityksen SaaS-palveluiden toimintavarmuutta saatiin parannettua merkittävästi.

Avainsanat	SaaS, toimintavarmuus, varmuuskopiointi, vikasietoisuus, saatavuus, palautuminen

Metropolia

Abstract

Author Title Number of Pages Date	Antti Vitikainen Research and development of the reliability of SaaS services 55 pages + 10 appendices 11 April 2012
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Data Networks
Instructor	Janne Salonen, Principal Lecturer

The objective of this thesis was to map the SaaS service reliability of TypingMaster Finland Inc, after which it was to be improved as seen fit. The main goal was to improve the redundancy and availability of the services.

Research was done to find out the main concerns regarding SaaS reliability. At a very early stage it became clear that data integrity was a major issue. Therefore a majority of this thesis deals with data backups and recovery.

The practical portion of the thesis began with mapping the earlier backup system, which was discovered to be insufficient. Afterwards the backup system was rebuilt using the resources available. Furthermore, a disaster recovery system was also built for the SaaS services mainly with MySQL replication, which utilized both Cold Standby and Hot Standby ideologies. The main principle was that both of the SaaS servers would have identical configurations, software and data, yet each service would be active at only one of the servers at any given time.

In addition to the replication, a backup system was built for the daily backup and storing of the service data. Archiving was also taken into consideration. Since customer data was also discovered to be business critical, the same backup system was implemented on the company's internal services as well.

After all the changes were made, possible threats to the company's SaaS services were mapped. Preparation for disaster recovery was also addressed, enabling a simple recovery procedure if needed.

Since the reliability of the company's SaaS services was noted to have vastly improved, the goals of this thesis were met. Furthermore, no animals were harmed in the making of this thesis.

Keywords	SaaS, reliability, backup, redundancy, availability, disaster recovery

Sisällys

Lyhenteet ja käsitteet

Apache Tomcat	Ohjelmisto, joka sisältää Apache-webbipalvelimen sekä Tomcat Servlet Containerin. Mahdollistaa Java-ohjelmien ajamisen webbiyhteyden yli.
Bash	Bourne Again Shell, yleisesti Unix-tyyppisissä käyttöjärjestelmissä käytössä oleva komentotulkki, mahdollistaa esimerkiksi skriptien ajon.
Batch	Skriptitiedosto, joka voidaan ajaa DOS- tai Microsoft Windows -käyttöjärjestelmissä.
cron	Unix-pohjaisille käyttöjärjestelmille tehty ajastuspalvelu, jolla voidaan ajaa komentoja, skriptejä ja ohjelmia haluttuihin kellonaikoihin.
DBMS	Database Management System, tietokannan hallintajärjestelmä eli ohjelmisto, jonka avulla voidaan hallita tietokantoja.
DDoS	Distributed Denial of Service, hajautettu palvelunestohyökkäys eli hyökkäys, joka tulee järjestelmällisesti useasta eri lähteestä samanaikaisesti.
DNS	Domain Name System, Internetin nimipalvelujärjestelmä, joka muuntaa verkkotunnuksia IP-osoitteksi.
Domain	Verkkotunnus, joka mahdollistaa palvelimiin ja koneisiin viittaamisen ilman niiden IP-osoitetta.
DoS	Denial of Service, palvelunestohyökkäys, jonka avulla pyritään lamaannuttamaan esimerkiksi webissä toimiva palvelu.

Downtime	Termi, jolla kuvataan aikaväliä, jonka ajan järjestelmä tai palvelu on tavoitettamattomissa.
Dump	Tietokannoista puhuttaessa eräänlainen varmuuskopio, joka sisältää tietokannan rakenteen ja yleensä myös datan.
Failover	Palvelun (automaattinen) siirtyminen järjestelmästä toiseen käyttökatkoksen sattuessa.
FTP	File Transfer Protocol, yksinkertainen tiedostonsiirtoprotokolla.
IaaS	Infrastructure as a Service, pilvipalvelumalli, jossa tarjotaan virtuaalipalvelimia.
iptables	Linux-käyttöjärjestelmiin sisäänrakennettu palomuuri.
iSCSI	Internet Small Computer System Interface, kiintolevy tai datavarasto, jota käytetään IP-verkon yli SCSI-komentojen avulla.
Java	Oliopohjainen ohjelmointikieli ja ohjelmien ajoalusta.
Linux	Linux-ydintä käyttävä Unixin kaltainen käyttöjärjestelmä.
LMS	Learning Management System, oppimisen hallintajärjestelmä eli ohjelmisto, joka mahdollistaa hallitun ja dokumentoidun opiskelun.
Load Balancer	Kuormantasaaja, joka jakaa verkkoliikenteen määritellysti kahden tai useamman laitteen välillä.
Managed Hosting	Palvelinvuokrausmalli, jossa asiakas saa haltuunsa koko palvelimen, jonka laitteistoa on mahdollista muokata tarpeiden mukaiseksi.

Middleware	Ohjelmistot, jotka mahdollistavat käyttäjäohjelmien ja käyttöjärjestelmän kommunikoinnin keskenään, esimerkiksi ajurit.
Multitenanttisuus	Pilvipalveluihin perustuva ohjelmistoperiaate, joka mahdollistaa yhden ohjelmistoinstanssin jakamisen useaksi eri käyttäjäsessioksi niin, että niiden data pysyy erillään.
MySQL	Suosittu relaatiotietokantaohjelmisto (RDBMS).
NAS	Network-attached Storage, tiedostopalvelin, joka toimii usein IP-verkon yli käytettävänä kiintolevynä.
NTBackup	Windows-käyttöjärjestelmien mukana tuleva varmuuskopiointiohjelma.
PaaS	Platform as a Service, pilvipalvelumalli, jossa ohjelmistoille tarjotaan ajoalusta, mutta itse palvelimeen ei ole käyttäjällä pääsyä.
RAID	Redundant Array of Independent Disks, tekniikka, jolla useampi fyysinen kiintolevy voidaan yhdistää yhdeksi loogiseksi, parantaen näin järjestelmän vikasietoisuutta.
RAID Controller	Fyysinen laite, jonka avulla RAID-levypakat usein toteutetaan.
Redundanssi	Periaate, jolla tarkoitetaan rinnakkaisia järjestelmiä, jossa varajärjestelmä odottaa pääjärjestelmän kaatumista.
Replikointi	Järjestelmän kahdentaminen tai peilaaminen eli reaaliaikainen kopiointi, yleisessä käytössä muun muassa tietokannoissa.

Robustius	Järjestelmän kyky jatkaa toimintaansa virheistä huolimatta, tähtää mahdollisimman yksinkertaiseen kokonaisuuteen.
SaaS	Software as a Service, pilvipalvelumalli, jossa ohjelmistoa tarjotaan palveluna eikä tuotteena.
SCORM	Shareable Content Object Reference Model, joukko digitaalisen oppimateriaalin standardeja, jota muun muassa LMS noudattaa.
Skripti	Kokoelma komentorivikäskyjä, joka voi parhaillaan muistuttaa jo tietokoneohjelmaa.
SLA	Service Level Agreement, palveluntarjoajan kanssa tehtävä sopimus, jossa määritellään palvelun tarkka kuvaus ja sisältö.
SQL	Structured Query Language, suosittu kyselykieli, jolla relaatiotietokantoihin voi tehdä hakuja, muutoksia ja lisäyksiä.
SSH	Secure Shell, salattuun tietoliikenteeseen tarkoitettu protokolla, jolla voidaan suojata esimerkiksi FTP- ja HTTP-liikennettä.
SVN	Subversion, versionhallintajärjestelmä, jonka tarkoituksena on ohjelmistojen lähdekoodin muokkaaminen hajautetusti tietoverkon yli.
Task Scheduler	Windows-käyttöjärjestelmissä mukana tuleva ajastuspalvelu, jolla voidaan ajaa komentoja, skriptejä ja ohjelmia haluttuihin kellonaikoihin.
Ubuntu	Debian-Linux-jakeluun perustuva käyttöjärjestelmä.

Uptime	Termi, jolla kuvataan aikaväliä, jonka ajan järjestelmä tai palvelu on saatavilla, kuvataan usein prosentteina.
Windows Server 2003	Microsoftin vuonna 2003 julkaisema palvelinkäyttöjärjestelmä.
Virtual Host	Tekniikka, jossa useita Domain-nimiä voidaan ylläpitää samalla palvelimella.
Virtualisointi	Tekniikka, jossa fyysiset järjestelmäresurssit jaetaan loogisiksi resursseiksi, mahdollistaa esimerkiksi virtuaalikoneiden (VPS) luomisen.
VPS	Virtual Private Server, palvelinvuokrausmalli, jossa asiakas saa käyttöönsä virtuaalipalvelimen, joka usein käyttää samoja fyysisiä resursseja muiden vastaavien kanssa.

1 Johdanto

Ohjelmistopalvelut kasvattavat suosiotaan. Tuotteena myytävien ohjelmistojen suosio on hiipumassa, jonka takia yhä useammat yritykset ja kehittäjät ovat ryhtyneet trendin mukaisesti tarjoamaan ohjelmistojaan palveluina. Tämän mahdollistaa tietysti Internet. Tällöin kuvaan tulee mukaan käsite SaaS, Software as a Service. Sitä käytetään terminä paljon, mutta liian monet yritykset keskittyvät vain sen tarjoamaan tuotteen imagonkohotukseen unohtaen vääjäämättä kasvavan vastuun.

SaaS:sta puhuttaessa merkittävä paino siirtyy palvelun jatkuvalle toimintavarmuudelle. Se luo aivan uudenlaisia teknisiä sekä hallinnollisia haasteita, joihin täytyy vastata ja jotka täytyy tiedostaa jo hyvissä ajoin. Tällöin ei enää riitä, että ohjelmisto luodaan, testataan ja myydään, vaan sen koko toimintainfrastruktuuria täytyy tukea. Pelkkä reaktiivinen tuki ei riitä, vaan sen täytyy olla aktiivista.

Tämä insinöörityö pureutuu ohjelmistopalvelujen toimintavarmuuteen. Työssä käsitellään asiaa sekä teoreettiselta kannalta, että myös teknistä käytännön toteutusta ajatellen. Painopisteenä ei ole kysymys ”miksi”, vaan pikemminkin ”miten”. Vaikka työssä kartoitetaankin palvelujen eri ominaisuuksien tärkeyksiä, ei siinä juurikaan keskitytä SaaS-palveluihin liiketoiminnan kannalta, vaan puhtaasti teknisestä näkökulmasta. Lisäksi, koska SaaS:n toimintavarmuus on aiheena niin laaja, keskitytään työssä yksinomaan yrityksellä käytössä oleviin ratkaisuihin. Työ jakaantuu teoriaosuuteen sekä yritykselle tehtyyn kartoitus- ja parannusosuuteen.

Insinöörityön toimeksiantaja on ohjelmistoyritys TypingMaster Finland Oy. Yritys on vuodesta 1992 asti toiminut suomalainen ohjelmistokehittäjä, joka tekee ja myy kymmensormitekniikan ohjelmistoja ja kursseja. Suurin osa yrityksen myynnistä koostuu viennistä ulkomaille, erityisesti Yhdysvaltoihin. Tuotteita yrityksellä on useita, mutta erityisesti suosiotaan kasvattavat SaaS-toimintamallilla toteutetut Internet-pohjaiset ohjelmistot. Osa palveluista on maksullisia ja osa ei. Insinöörityössä keskitytään näiden ohjelmistopalvelujen toimintavarmuuteen ja siihen, miten yrityksen IT-infrastruktuuri valjastetaan tukemaan tätä. Erityisen suuri painopiste annetaan palvelujen datan käsittelylle, suojaamiselle ja varastoinnille sen business-kriittisyyden takia.

Työn kirjoitushetkellä osa yrityksen SaaS-palvelujen parannuksista on jo tehty. Ensin työssä käydään läpi näiden palvelujen IT-infrastruktuurin ja toimintavarmuuden aiempi tilanne, minkä jälkeen tehdään suunnitelma parannuksista ja toteutetaan ne. Mikäli puutteita ja parannusehdotuksia tulee vielä tämän jälkeen esille, pyritään niitä mahdollisuuksien mukaan vielä toteuttamaan. Lisäksi työssä kartoitetaan palvelujen nykytilanteeseen kohdistuvia riskejä ja uhkia ja sitä, miten niihin varaudutaan.

Koska työ sisältää yrityksen käytössä olevaa tietoa ja skriptejä, on niistä tarkoituksellisesti poistettu kaikki kriittinen informaatio kuten IP-osoitteet, portit, käyttäjätunnukset ja salasanat.

2 Software as a Service

Software as a Service (SaaS) on vielä uudehko termi, jota käytetään laajalti, mutta ymmärretään harvemmin. Se on osa pilvipalvelu-ajatusmaailmaa, jonka pääperiaatteena on palvelun tarjoaminen asiakkaalle ilman tuotteen omistajuuden vaihtoa. Myös muita 'as a Service' -palvelumalleja on olemassa, mutta suosituin on SaaS eli vapaasti suomennettuna 'ohjelmisto palveluna', sillä sen asiakasryhmään kuuluvat pääsääntöisesti aivan tavalliset kuluttajat (vrt. IaaS, PaaS). Käytännössä SaaS tarkoittaa sitä, että ohjelmistoa ei myydä asiakkaalle itsenäisenä tuotteena, jonka kokonaisvaltainen käyttö kaikkine hyötyineen ja haittoineen on oston jälkeen asiakkaan vastuulla. Tällöin siirrytään siis tuote-ajattelusta palvelu-ajatteluun. [1.] Yritykselle tämä luo aivan uusia haasteita ja ennen kaikkea lisää vastuuta, sillä sen vastuulle jäävät palvelun asennus-, ylläpito- ja huoltotoimet [2, s. 10].

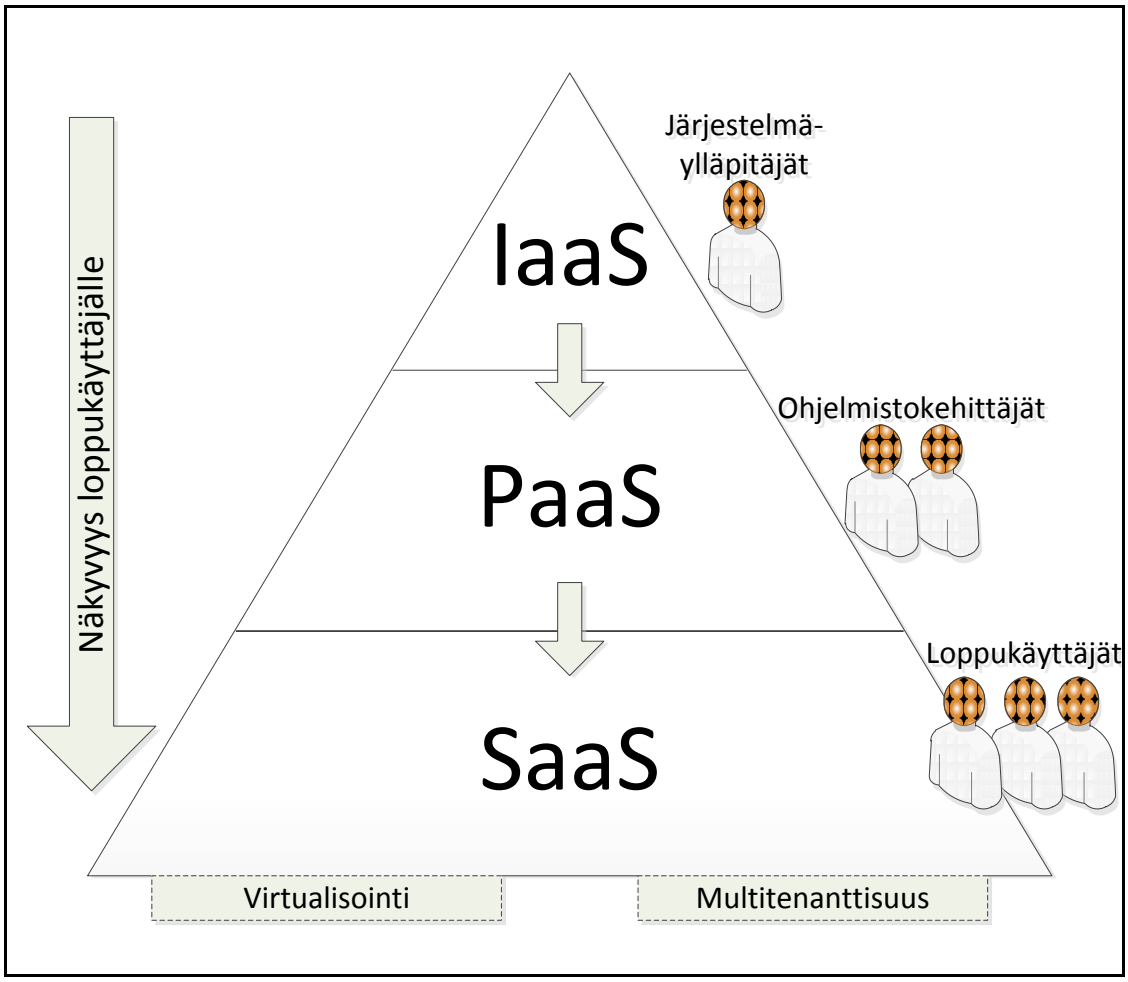
Tyypillisesti SaaS-palvelua käytetään web-selaimella verkon yli. Periaatteena on, että käyttäjä ottaa yhteyden jossakin muualla toimivaan pilveen ja voi tämän jälkeen vapaasti käyttää tälle myönnettyjä palveluja ja resursseja. SaaS:n tapauksessa tämä tarkoittaa yhtä tai useampaa ohjelmistoa. Nämä ohjelmistot toimivat keskitetysti palvelun-tarjoajan palvelimilla, ja yksi sovellusinstanssi voi palvella useaa asiakasta samanaikaisesti. Tämä mahdollistaa palvelun merkittävän skaalautuvuuden sekä tehokkaan moni-

toroinnin. [2, s. 10.] Esimerkiksi erilaiset sähköpostipalvelut, kuten suosittu Gmail, noudattavat SaaS-ajattelua.

Käyttäjän näkökulmasta SaaS on ennen kaikkea kustannustehokas ratkaisu. Asiakkaan ei tarvitse ostaa kokonaista tuotepakettia, vaan käyttöön voidaan ottaa vain ne ominaisuudet, joita tarvitaan. Tyypillisesti SaaS-palvelua myydäänkin ns. 'pay as you go'- eli kestotilausperiaatteella, joten myös tuotteesta irtisanoutuminen on mahdollista. Asiakas säästää hallintakulujen lisäksi hankinnan kokonaiskustannuksissa, sillä palvelun käyttöönotto ja muokkaus omiin tarpeisiin ovat yksinkertaisia toimenpiteitä. Lisäksi palvelun säännöllinen päivitys on helpompaa, eikä vaadi asiakkaalta mitään toimenpiteitä. Myös helppo käytettävyys tulee ottaa huomioon, sillä tyypillisesti SaaS-palvelua voidaan käyttää lähes mistä vain, mikäli tähän on lupa. [3.]

SaaS eroaa muista pilvipalvelumalleista siten, että se on usein ainut palvelumalli, jonka kanssa loppukäyttäjä pääsee tekemisiin (kuva 1). IaaS (Infrastructure as a Service) tarjoaa yrityksille räätälöitävän virtuaalitietokoneen, jonka päällä omia järjestelmiä ja ohjelmistoja voi ajaa. PaaS (Platform as a Service) taas mahdollistaa ohjelmistojen ajon jo valmiissa alustassa. SaaS taas on valmis ohjelma, johon loppukäyttäjät pääsevät helposti käsiksi. [4.]

Kaikkien pilvipalvelumallien (IaaS, PaaS ja SaaS) kulmakivinä ovat virtualisointi ja multitenanttisuus. Virtualisoinnilla tarkoitetaan sitä, että käytössä olevat fyysiset resurssit voidaan jakaa useiksi loogisiksi resursseiksi. Palvelun käyttäjällä ei ole siis koskaan täyttä hallintaoikeutta koko infrastruktuuriin. [5, s. 78.]



Kuva 1. Pilvipalvelumallit ja niiden asiakaskunnat

Multitenanttisuus (multi tenant = monta vuokralaista) taas on ohjelmistotermi, jolla tarkoitetaan yhden ohjelmiston kykyä jakaa jokaiselle asiakkaalle oma sessionsa ja täten skaalautua tehokkaasti. Tällöin vaikka kaikki asiakkaat käyttävät samaa ohjelmistoa, pidetään asiakaskohtainen data erillään toisistaan. Koska kaikkia asiakkaita palvelee yksi ja sama sovellus, heijastuvat mahdolliset sovellukseen tehdyt muutokset heti kaikille asiakkaille. Tämä mahdollistaa päivitysten ja korjausten nopean implementoinnin. Multitenanttisuus on erityisesti SaaS-palvelujen kulmakivi, sillä sen avulla ohjelmisto voi palvella useaa asiakasta samanaikaisesti. [6.]

Vaikka SaaS tarjoaa oikein tehtynä merkittäviä hyötyjä sekä asiakkaille että yrityksille, luo se myös haasteita, jotka palveluntarjoajan on välttämätöntä tiedostaa. Suuri osa SaaS-ajattelusta perustuu siihen, että asiakkaat luottavat palvelun saatavuuteen ja toimivuuteen, eikä tätä luottamusta ole syytä rikkoa. Usein myös asiakkaan tuottama data sekä työpanos sijaitsevat palveluntarjoajan palvelimilla, minkä ansiosta myös tur-

vallisuuteen täytyy panostaa. [7.] Todella merkittävä paino siirtyy siis palvelun toimintavarmuudelle.

3 Toimintavarmuus

Toimintavarmuus on käsite, joka voi IT-järjestelmistä puhuttaessa tarkoittaa useita eri asioita. Sille ei ole varsinaista englanninkielistä käännöstä, mutta sen alle voidaan summata monta vastaavaa jo alalla vakiintuneempaa termiä. Näitä ovat esimerkiksi luotettavuus, redundanssi, vikasietoisuus, saatavuus ja failover, jotka kaikki omalta osaltaan täydentävät järjestelmän toimintavarmuutta. IT-ohjelmistoista puhuttaessa toimintavarmuudella kuvataan ohjelmiston kykyä toimia ennaltamäärätysti ilman virheitä tai ongelmia [8].

SaaS-palvelujen toimintavarmuus on monimutkainen käsite, sillä palvelut ja järjestelmät jotka niitä pyörittävät voivat olla todella erilaisia. Palvelu voi pyöriä esimerkiksi PaaS- tai IaaS-järjestelmässä, jolloin palvelun infrastruktuuri on aivan erilainen. Lisäksi tyypillinen SaaS-palvelu koostuu useista tasoista (verkkoinfrastruktuuri, laitteisto, palvelinohjelmisto, tuoteohjelmisto), minkä takia yhdenkin komponentin rikkoutuminen voi samalla rikkoa koko palvelun. Oli syy rikkoutumiseen mikä tahansa, loppukäyttäjän näkökulmasta palvelu ei vain tällöin toimi. Tämä taas ei ole hyvä asia, sillä toimintavarmuus on tärkein aspekti, jota ihmiset SaaS:ltä haluavat [9, s. 17].

SaaS-palvelujen toimintavarmuutta voidaan edistää monella eri tavalla. Olennaisin lähtökohta on, että sekä infrastruktuurin ja ohjelmiston tulisi olla mahdollisimman luotettavaa. Toisin sanoen siis jo ennen palvelun myyntiä tulisi niille suorittaa kattavat testaukset, jonka avulla mahdolliset bugit saadaan paikannettua ja korjattua. On tärkeää tarkistaa ohjelmiston resurssikäyttö eli se, miten paljon se käyttää prosessointitehoja, levyä ja muistia. Muistin käytön seuraaminen on erityisen tärkeää, sillä parhaimmissakin ohjelmistoissa voi usein ilmetä muistivuotoja. Varsinkin multitenanttisissa SaaS-ohjelmistoissa tämä voi kaataa palvelun hyvin nopeasti. [5, s. 227-228.]

Kun palvelu toimii halutusti, voidaan sen toimintavarmuutta lähteä edistämään monella eri tavalla. Kaikki palvelut ovat erilaisia, joten yhtä ja oikeaa tapaa ei tähän ole. Siitä

huolimatta joitain yleisiä suosituksia ja ohjesääntöjä olisi hyvä seurata. Pahimmillaan toimintavarmuutta parantamaan luotu järjestelmä voi vaikuttaa palveluun juuri päinvastoin, jos se on toteutettu huolimattomasti. Lähtökohtana on, että ensin täytyy tuntea palvelun ohjelmisto, infrastruktuuri ja käyttö. Tämän pohjalta voidaan hahmotella mahdolliset uhat ja miten niihin vastataan. Kun tämäkin on selvillä, voidaan järjestelmän toimintavarmuutta lähteä parantamaan. Abstraktilla tasolla voidaan toimintavarmuus jakaa kahteen periaatteeseen:

1. Yksinkertaisuus ja robustius: Minimoidaan virheiden mahdollisuus.
2. Vikasietoisuus ja redundanssi: Luodaan järjestelmä, jonka avulla palvelun toimintaa voidaan jatkaa virheen sattuessa. [5, s. 34.]

Robustiudella (robust = sitkeä) tarkoitetaan järjestelmän kykyä jatkaa toimintaansa virheestä huolimatta. Tämä on vahvasti kytköksissä yksinkertaisuuteen, sillä ominaisuudet joita ei tarvita, tulisi poistaa kokonaan. Tällä tavoin minimoidaan mahdolliset virheiden aiheuttajat. Kuten Albert Einstein sanoi: ”Kaikesta tulisi tehdä niin yksinkertaista kuin mahdollista, mutta ei sen yksinkertaisempaa.” [10.]

Redundanssilla (redundant = ylimääräinen) taas tarkoitetaan yleensä rinnakkaisia järjestelmiä. Mikäli ensimmäinen järjestelmä kohtaa virheen ja menee alas, voi toinen järjestelmä jatkaa siitä. Tätä ajattelua voidaan soveltaa lähes kaikissa SaaS-palvelun komponenteissa kuten esimerkiksi verkkoyhteydessä, palvelimissa, DNS:ssä tai itse ohjelmistossa. [11.]

Yksinkertaisuus ja redundanssi ovat usein ristiriitaisia periaatteita, sillä redundanssi pyrkii lisäämään järjestelmään ylimääräisiä resursseja, tehden siitä samalla monimutkaisemman. Tällöin yksinkertaisuuden periaate ei toteudu, sillä jokainen uusi komponentti luo uusia mahdollisuuksia virheille. [5, s. 35.] Näiden kahden periaatteen välille tulisikin saada tasapaino, jonka löytämiseen tarvitaan kokemusta ja palvelun tuntemusta.

Erilaisia menetelmiä toimintavarman SaaS-palvelun luomiseen on lukemattomia. Suurimmassa roolissa on kuitenkin lähes aina palvelun data, joka on korvaamatonta. Tä-

män takia varmuuskopiointi on ehkä olennaisin osa palvelujen toimintavarmuutta. [12, s. 689–693.]

3.1 Varmennus

Varmennus on redundanssiin nojaava prosessi, jolla tarkoitetaan yleensä replikaatiota tai peilausta. Ylivoimaisesti yleisin käytössä oleva varmennuskeino on RAID (Redundant Array of Independent Disks), jonka avulla fyysisesti erillisistä kiintolevyistä luodaan yksi tai useampi looginen levy. Tämä voidaan toteuttaa usealla eri tavalla (RAID-0, RAID-1, RAID-3, RAID-5 ja RAID-6). Yleisin tapa on RAID-1 eli peilaus. Tällöin kaksi kiintolevyä sisältää tarkalleen saman datan, ja toisen hajotessa voidaan toinen ottaa käyttöön ilman datahäviötä. RAID-1 onkin täydellinen esimerkki redundanssista. [5, s. 109-117.]

Varmennusta voidaan toteuttaa myös muuten kuin laitteistotasolla. Esimerkiksi SQL-replikaatio on yleinen prosessi, jonka avulla kaksi SQL-tietokantaa synkronoidaan keskenään. Yleensä tämä tapahtuu yksisuuntaisesti. SaaS-palveluissa tietokantasynkronointi on käytössä laajalti. [13.]

3.2 Varmuuskopiointi

Kuten on jo useasti todettu, varmuuskopiointi on erityisesti SaaS-palveluille tärkeää. Asiakas luottaa datansa palveluntarjoajan haltuun, jolloin siitä täytyy myös pitää huolta. On aina mahdollista, että pääasiallinen data korruptoituu tai tuhoutuu, jonka jälkeen täytyy palauttaa datan varmuuskopio. Käytännössä varmuuskopio on toissijainen kopio päädatasta, ja sellaisena se pysyy, kunnes se täytyy palauttaa [12, s. 696]. Kuten kirjassa ’High Availability & Disaster Recovery’ todetaan, varmuuskopioista ei ole mitään hyötyä, jos niitä ei voida palauttaa. Kaikkiin varmuuskopioihin pätee tämä sama sääntö. Palautumisen onnistumiseksi varmuuskopioidun datan tulee olla

1. ehjää ja oikeellista
2. ajantasaista
3. palautettavissa ennaltamäärätyn aikataulun mukaisesti. [5, s. 284.]

3.2.1 Mitä?

Se mitä varmuuskopioidaan, riippuu täysin yrityksestä ja sen palveluista. Yleensä varmuuskopioitava data voidaan kumminkin SaaS-palveluissa jakaa kolmeen eri luokkaan. Nämä ovat immateriaaliomaisuus, asiakasdata ja käyttäjädata. Immateriaaliomaisuudella tarkoitetaan kaikkea yrityksen liiketoiminnalle kriittistä dataa kuten esimerkiksi ohjelmien lähdekoodeja ja verkkolevytiedostoja. Asiakasdata taas sisältää yrityksen asiakkaiden henkilötiedot, laskutustiedot ja esimerkiksi käyttöoikeustiedot. Käyttäjädata eroaa asiakasdatasta siten, että se on asiakkaan itse tuottamaa dataa yrityksen SaaS-palvelussa. Se on siis luonnollisesti kaikkein kriittisin varmuuskopioitava. [12, s. 698.]

Teknisellä tasolla varmuuskopioitavaa dataa on kolmea eri tyyppiä. Nämä ovat järjestelmävarmuuskopiot, tiedostovarmuuskopiot sekä tietokantavarmuuskopiot. Tyypillisesti asiakas- ja käyttäjädatasta otetaan tietokantavarmuuskopiot (voivat olla myös tiedostoja), kun taas immateriaaliomaisuudella tarkoitetaan yleensä tiedostoja. Järjestelmävarmuuskopioita ei useimmiten SaaS-palveluista oteta. [5, s. 283.]

Tietokantavarmuuskopiot ovat käytännössä lähes aina dumppeja eli kopioita halutuista tietokannoista. Ne voidaan toteuttaa monella eri tavalla, mutta yleisimmät tietokantojen hallintajärjestelmät (DBMS, Database Management System) sisältävät työkalun tätä varten. Esimerkiksi suositussa MySQL:ssä on mukana työkalu MySQLDump. [12, s. 391–396.]

Tiedostovarmuuskopiot eivät aina tarkoita täydellistä kopiota kaikista tiedostoista, vaan ne voidaan jakaa eri tasoihin. Yleisimmät tasot ovat

- full (täydellinen kopio kaikesta)
- inkrementaalinen (kopio kaikesta mikä on muuttunut edellisen varmuuskopion jälkeen)
- differentiaalinen (kopio kaikesta mikä on muuttunut edellisen täydellisen varmuuskopion jälkeen). [12, s. 28.]

Kuten päätellä saattaa, ei ole tehokasta ottaa päivittäin täydellistä varmuuskopiota kaikista tiedostoista. Se riippuu tiedostojen määrästä ja luonteesta, millaisia varmuuskopioista niistä tulisi ottaa. Ehkäpä yleisin toimintatapa on ottaa viikoittain täydellinen varmuuskopio ja päivittäin differentiaalinen. [12, s. 30–32.]

3.2.2 Miten?

Varmuuskopioita voi ottaa lukemattomilla eri tavoilla riippuen laitteistosta, infrastruktuurista, käyttöjärjestelmästä ja käytetyistä ohjelmistoista. Maailmassa on olemassa vähintään satoja kaupallisia ratkaisuja, joiden avulla pystytään varmuuskopioimaan mitä vain [12, s. 203–204]. Varmuuskopiointi jakaantuu kumminkin aina kahteen prosessiin: varmuuskopion generointiin ja sen siirtoon muualle talteen. Täten myöskään itse rakennettua varmuuskopiointijärjestelmää ei ole hankalaa tehdä. [12, s. 59.]

Kuten edellisessä kappaleessa todettiin, tehdään useimmat tietokantavarmuuskopiot niihin integroiduilla työkaluilla. Tämä johtuu ensinnäkin erilaisten tietokantojen hallintajärjestelmien eroavaisuuksista, ja toiseksi tietokantojen monimutkaisuudesta. Tietokannat muuttuvat jatkuvasti ja ne voivat olla todella massiivisia. Tämän takia useimmat valmiit työkalut toimivat yhteistyössä tietokantamoottorin kanssa, jolloin ne pystyvät esimerkiksi hetkellisesti pysäyttämään tietokannan muokkauksen varmuuskopioinnin ajaksi. [12, s. 392–395.]

Tiedostojen varmuuskopiointi voidaan tehdä monella eri tavalla. Esimerkiksi Linux-pohjaisissa käyttöjärjestelmissä on jo valmiiksi olemassa cpio, dd, dump, rsync ja tar -ohjelmistot, joilla kaikilla voidaan tehdä jonkinlaista varmuuskopiointia. Vanhemmissa Windows-käyttöjärjestelmissä on taas NTBackup-ohjelma, jolla levyjen, hakemistojen ja tiedostojen varmuuskopiointi toimii vaivattomasti. Uudemmissa Windows-käyttöjärjestelmissä (Windows 7, Windows Server 2008) taas on valmis Backup & Restore -ohjelma. [12, s. 59–63.]

Varmuuskopiointiprosessin käynnistys täytyy yleensä automatisoida. Mikäli käytössä on jotain muuta kuin maksullinen varmuuskopiointiohjelmisto, toteutetaan tämä yleensä joko Cron-ohjelmistolla (Linux) [14] tai Task Schedulerilla (Windows) [15]. Molempien avulla prosessi voidaan käytnnistää jopa minuutin välein.

Valmis varmuuskopio täytyy aina siirtää talteen. Tämä voidaan tehdä esimerkiksi fyysiselle medialle, kuten muistitikulle tai levykasetille. Yleensä kuitenkin se siirretään verkon yli suojatun yhteyden avulla (SSH, FTP, sFTP, SCP).

3.2.3 Milloin?

Ei ole yhdentekevää, milloin varmuuskopiot otetaan. Minkä useammin prosessi suoritetaan, sen ajantasaisempaa varmuuskopioitu data on. Kääntöpuolena on se, että tämä on usein tarpeetonta ja kuormittaa laitteistoa suotta. Riippuu täysin yrityksestä ja sen palveluista, milloin ja miten usein on optimaalista tehdä varmuuskopio. Ohjesääntönä voidaan kuitenkin sanoa, että palvelua pitäisi häiritä mahdollisimman vähän. Käytännössä tämä tarkoittaa usein sitä, että varmuuskopio tulisi tehdä yöllä, jolloin käyttäjiä on mahdollisimman vähän. SaaS-palveluissa tämä ei kuitenkaan aina ole mahdollista, sillä palveluja käytetään usein ympäri maapalloa. [12, s. 27–33.]

On tärkeää suunnitella, miten kauan varmuuskopioita säilytetään. Nykyään data vanhenee nopeasti, joten kaikkien varmuuskopioiden säilyttäminen vuosikausia on turhaa. Tätä vastoin arkistointia on useimmissa tapauksissa hyvä harjoittaa. Arkistoitu data on eräänlainen varmuuskopion varmuuskopio. Se ei ole enää palautumisen kannalta oleellista, mutta erinäisistä syistä sitä voidaan vielä joskus tarvita. [12, s. 696–697.]

3.3 Monitorointi ja valvonta

Toimintavarmuuden takaamiseksi täytyy palveluja monitoroida. Tämä ei kuitenkaan tarkoita vain palvelun toimintaa, vaan myös sen taustalla toimivia komponentteja, kuten esimerkiksi varmuuskopiointijärjestelmää. Monitoroinnin ja valvonnan pääasiallinen rooli on virheiden, puutteiden ja ongelmien havaitseminen ennen kuin ne vaikuttavat palvelun saatavuuteen. Aina ei ole mahdollista puuttua tilanteeseen ennen palvelukatkosta, mutta tärkeintä on sentään saada virheestä ilmoitus. Monitoroinnin ja virheilmoitusten täytyy siis olla täysin automaattisia prosesseja. On erityisen tärkeää, että vakavat palvelukatkokset tunnistetaan heti ja että niistä tulee ilmoitus. Valmista palautusprosessia ei yleensä voida nopeuttaa, mutta aikaväliä palvelun kaatumisen ja sen tiedostamisen välillä voidaan aina pienentää. [5, s. 284–286.]

Monitorointi voidaan toteuttaa monella tavalla, riippuen siitä mitä monitoroidaan. SaaS-palvelun saatavuuden valvontaa varten on olemassa useita valmiita palveluja, jotka käytännössä tarkistavat tietyin väliajoin onko palvelu tavoitettavissa. Mikäli näin ei ole, lähetetään sähköposti-ilmoitus ennaltamäärättyyn osoitteeseen. Monitorointia voidaan

tehdä myös järjestelmän sisällä. Usein palvelimissa valvotaan jatkuvasti esimerkiksi niiden lämpötiloja, muistin käyttöä, verkkoliikennettä ja kiintolevyjen tilaa. Näiden toteutusta varten on olemassa monia eri ohjelmistoja. Useimmat käyttöjärjestelmät pitävät myös itse erilaisia lokitiedostoja niiden toiminnasta. [16.]

Myös varmuuskopiointijärjestelmän toimintaa on hyvä valvoa, jotta varmuuskopioinnin tiedetään toimivan [12, s. 48]. Lukemattomat kauhutarinat muistuttavat siitä, mitä tapahtuu, jos palvelu tuhoutuu eivätkä varmuuskopiot toimikaan [17].

3.4 Muut tekijät

Datan turvaaminen ei ole tietenkään ainut asia, joka vaikuttaa SaaS-palveluiden toimintavarmuuteen. Palvelujen verkkoinfrastruktuuri on täysin oma lukunsa, eikä sen toimintaa juurikaan käydä tässä työssä läpi. Tämä johtuu pääasiassa siitä, että yrityksen SaaS-palvelujen verkkoinfrastruktuuri on kolmannen osapuolen vastuulla.

Eräs erityisen tärkeä toimintavarmuuteen vaikuttava palvelu on DNS eli Domain Name System. Käytännössä se on Internetissä sijaitseva rekisteri siitä, mikä IP-osoite vastaa mitäkin Domain-nimeä. Se on siis tärkeä komponentti, jonka toiminta täytyy olla erityisen varmaa. Lähes aina käytetäänkin kolmannen osapuolen DNS-palveluja. Toimintavarmuuden kannalta olennaisinta on kuitenkin se, että DNS:än avulla voidaan palvelujen saatavuutta hallita ohjaamalla liikennettä eri palvelimiin. Mikäli palautuminen toiselle palvelimelle täytyy suorittaa, täytyy uudelleenohjaus sinne useimmiten suorittaa DNS:än avulla. Jotkin palveluntarjoajat tarjoavat myös palvelua, joka automaattisesti vaihtaa DNS:än osoittamaan toiselle palvelimelle, jos ensimmäiseen ei saada yhteyttä. [5, s. 271–273.]

4 Palvelukatkokset

SaaS-palvelujen pahimpia vihollisia ovat palvelukatkokset. Siinä missä lisenssipohjaisten asennettavien ohjelmistojen suurimpia ongelmia ovat perinteisesti olleet bugit, ei SaaS-palveluissa asia ole näin. Ohjelmistobugeja tulee aina. Asennettavissa ohjelmissa niiden korjaus ei ole mitenkään vaivaton prosessi, sillä korjauksen jakelu on vaivalloi-

ta. SaaS-pohjaisessa ohjelmistossa tästä on päästy eroon, sillä korjaukset voidaan tehdä keskitetysti lähes välittömästi [1]. Suurimman ongelman luo tätä vastoin palvelun saatavuus, sillä palvelujen täytyy olla aina saatavilla, jotta liiketoiminta olisi uskottavaa ja kannattavaa. SaaS-palvelujen saatavuus ei ole myöskään mikään yksinkertainen asia, vaan siihen vaikuttaa useita eri komponentteja.

SaaS-palveluihin kohdistuu erilaisia uhkia, virheitä, katastrofeja ja sitä kautta palvelukatkoksia. Osa katkoksista on lyhyitä, jopa niin lyhyitä ettei niitä ehdi edes huomata, ja osa taas sellaisia, että niistä palautumiseen tarvitaan manuaalinen palautusoperaatio. Näiden ääripäiden väliltä löytyy paljon monimutkaisempia tapauksia, mutta karkeasti sanottuna palvelukatkokset voidaan jakaa juuri näihin kahteen ryhmään. Lyhytkestoinen katkos kuuluu saatavuuden ja vikasietoisuuden piiriin, kun taas pitkäkestoinen vaatii palautusta. Myös syyt katkoksiin ovat moninaisia ja usein tapauskohtaisia. Onkin hyvä huomioida, että eri uhkia tai virheitä ei voida suoraan yhdistää tietyntyyppiseen katkokseen, vaan ne täytyy käsitellä eri asioina. [5, s. 21.]

4.1 Uhat

Mitä kompleksimpi järjestelmä on kyseessä, sen enemmän siihen kohdistuu riskejä. Tämä pitää paikkansa varsinkin SaaS-palveluiden kohdalla, sillä niiden toimintavarmuus on hyvin pitkälti verkkoliikenteen varassa. Erilaisia palveluun kohdistuvia uhkia voidaan luokitella todennäköisyyden ja vakavuuden mukaan. Kaikkeen ei tietenkään ole mahdollista varautua etukäteen, mutta tavallisimmat uhat ja virheet on aina hyvä tiedostaa. Garfinkel ja Spafford kirjassaan ’Practical UNIX & Internet Security’ luokittelevat IT-järjestelmiin kohdistuvat virheet yhdeksään eri kategoriaan [18, s. 545–546]. Alle on poimittu niistä SaaS-palveluille oleellisimmat.

- Käyttäjävirhe

  SaaS-palveluissa tämä ei ole yleistä, sillä käyttäjien pääsy eri toimintoihin on tavallisesti hyvin rajoitettua. Käyttäjä ei yleensä saa palvelua tavallisella käytöllä rikottua, vaikka niin haluaisikin. Vaikka käyttäjävirhe tapahtuisikin, koskee se yleensä käyttäjädataa, eikä siten ole itse palvelun kannalta vakava. Paras puolustautumiskeino näitä virheitä vastaan ovat varmuuskopiot.

- Ylläpitovirhe

  Mikäli palvelun ylläpitäjä tekee vahingossa pahan virheen, voivat seuraukset olla palvelun kannalta katastrofaaliset. Virhe voi kohdistua ohjelmistoon, järjestelmään tai infrastruktuuriin eli siis mihin vain, minne ylläpitäjällä on pääsy. Nämä virheet ovat vakavimmasta päästä, sillä ne saattavat pahimmillaan väliaikaisesti tuhota koko palvelun. Paras keino ylläpitovirheiden ehkäisyyn on huolellisuus ja dokumentointi.

- Laitteistovirhe

  Tämä on IT-järjestelmien yleisin uhka, johon täytyy varautua aina. Laitteistovirhe tarkoittaa yksinkertaisesti yhden tai useamman laitteen tai laitekomponentin hajoamista. SaaS-palvelujen kohdalla tämä tarkoittaa käytännössä aina palvelinta, jossa palveluja pidetään. On tärkeää huomata myös, että laitteistovirhe voidaan jakaa kahteen eri kategoriaan: kiintolevyvirheisiin ja järjestelmälaajuisiin virheisiin [5, s. 34].

- Kiintolevyvirhe

  Mikäli palvelimen kiintolevy menee rikki, on siinä oleva data käytännössä menetetty, ellei varmuuskopioita ole. Tämän takia todella usein käytetään vähintään RAID-levypakkoja. RAID voidaan toteuttaa monella eri tavalla. Yleisimpiä ovat lomitus (RAID-0), peilaus (RAID-1) ja pariteettidataa käyttävät RAID-5 ja RAID-6. Lisäksi käyttäjädatasta on hyvä olla varmuuskopiot myös muualla, mikäli koko levypakka hajoaa. Kiintolevyn hajoaminen ei yleensä vaikuta mitenkään muihin järjestelmäkomponentteihin, vaikka tekeekin palvelusta käyttökelvottoman.

- Järjestelmälaajuinen virhe

  Järjestelaajuisilla virheillä tarkoitetaan laitteistovirheitä, jotka tekevät koko järjestelmästä käyttökelvottoman. Näitä ovat esimerkiksi prosessori-, muisti- ja ilmastointivirheet, joiden seurauksena koko palvelininfrastruktuurille voi käydä huonosti. Valitettavan usein järjestelmälaajuiset virheet johtuvat ylläpitovirheistä tai ylläpitäjien huolimattomuudesta. Varoituksia mahdollisista uhista pitäisi tulla ja ne täytyy tiedostaa.

- Verkkovirhe

  Vaikka tätä ei kirjassa [18, s. 545-546] luetellekaan, on verkkovirhe SaaS-palvelujen kannalta todella huomattava ja todennäköinen uhka. Mikäli verkkoyhteys katkeaa missä vain käyttäjän ja palvelimen välillä, ei palve-

lun käyttö enää onnistu. Virheenä se on usein sellainen, ettei SaaS-palvelun tarjoaja voi asialle mitään. Verkkovirheiltä vältytään parhaiten erilaisilla redundanssiratkaisuilla, kuten esimerkiksi kahdennetulla verkkoyhteydellä.

- Ohjelmistovirhe

  Suojautuminen ohjelmistovirheiltä on hankalaa. Virhe voi tapahtua käyttöjärjestelmässä, tietokannassa, palvelinohjelmistossa tai itse SaaS-palvelussa, eikä sen paikantaminenkaan ole usein helppoa. Tämän takia on tärkeää, että kaikki kriittiset ohjelmistot pitäisivät lokitiedostoa.

- Luvaton murtautuminen, vandalismi ja hyökkäykset

  Nykymaailmassa yritykset ovat todella suosittuja kohteita virtuaalirikollisuudelle. Näiden murtautumisten ja hyökkäysten kautta pyritään usein vain haitantekoon, mutta joskus myös rahalliseen hyötyyn. Tällaiset murtautumiset ovat erittäin vakava uhka, sillä pahimmillaan murtautuja pystyy tuhoamaan koko palvelun. Vaikka data saataisiinkin palautettua, ei koskaan voida olla varma, mitä sille tapahtui murtautumisen aikana. Erityistä päänvaivaa SaaS-palveluille aiheuttavat palvelunestohyökkäykset (DoS attack). Näiltä suojaudutaan parhaiten pitämällä yllä hyvää tietoturvaa.

- Katastrofit

  Kaikkiin IT-järjestelmiin voi kohdistua erilaisia katastrofeja. Ne voivat olla joko luonnonkatastrofeja, tulipaloja tai mitä vain kuvitella saattaa. Tällöin koko järjestelmä tuhoutuu. Ainut keino tuhojen ehkäisemiseksi näissä tapauksissa on palvelujen hajautus fyysisesti. Eri laitteille hajauttaminenkaan ei aina riitä, vaan palvelimia on hyvä olla myös pitkien matkojen päässä. Tyypillisesti vähintään tuki- ja palveluverkko pidetään erillään. [12, s. 36.]

4.2 Saatavuus

IT-järjestelmien saatavuus määritellään seuraavasti:

> Saatavuus on järjestelmän ominaisuus suojautua lyhyiltä katkoksilta ja palautua niistä lyhyellä aikavälillä yleensä automaattisesti. [5, s. 22.]

Vaikka toimintavarmassa SaaS-järjestelmässä pyritäänkin jatkuvaan saatavuuteen, ei se käytännössä koskaan ole täysin mahdollista. On hyvä huomata, että jatkuva saatavuus tarkoittaa 100 %:n käytettävyysaikaa ja on siten terminä eri asia kuin pelkkä saatavuus. Jatkuva saatavuus onkin helposti yhdistettävissä vikasietoisuuteen eli siihen, ettei palvelukatkoksia yksinkertaisesti tapahdu. Oikeissa tilanteissa pelkästään tähän ei tietenkään voida turvautua. [5 s. 21–23.]

Kuten aikaisemmin todettiin, lyhytkestoiset ja vähäpätöiset katkokset kuuluvat saatavuuden piiriin. Ei ole väliä, tapahtuuko virhe ohjelmistossa, järjestelmässä, infrastruktuurissa tai ihmisen aikaansaamana. Palvelun täytyy palautua itsestään lyhyellä aikavälillä. SaaS-palveluissa tyypillisimpiä tällaisia tapauksia ovat Internet-yhteyden hetkellinen katkeaminen tai palvelimen ylikuormittunut prosessori. Käyttäjä saattaa joutua hetken odottamaan, mutta tyypillisesti sekuntien tai korkeintaan minuuttien kuluttua palvelu on taas täydessä toiminnassa.

Saatavuutta voidaan myös mitata. Kaava sen laskemiseen on [5, s. 24]:

$$Saatavuus = \frac{uptime}{uptime + downtime}$$

Tyypillisesti saatavuutta mitataan prosentteina. Usein eri palveluntarjoajat lupaavat jo SLA:ssa tietyn prosenttimäärän sille, kuinka hyvin palvelu on vähintään saatavilla esimerkiksi per kuukausi. Tyypillisesti tämä on vähintään 99,0 %.

4.3 Palautuminen

Palautuminen määritellään seuraavasti:

> Palautuminen on kyky jatkaa palveluiden toimintaa vakavan tai pitkäkestoisen katkoksen jälkeen, vaikkakin usein pienemmillä tehoilla. Palautumiseen yleensä liittyy manuaalinen prosessi. [5, s. 26.]

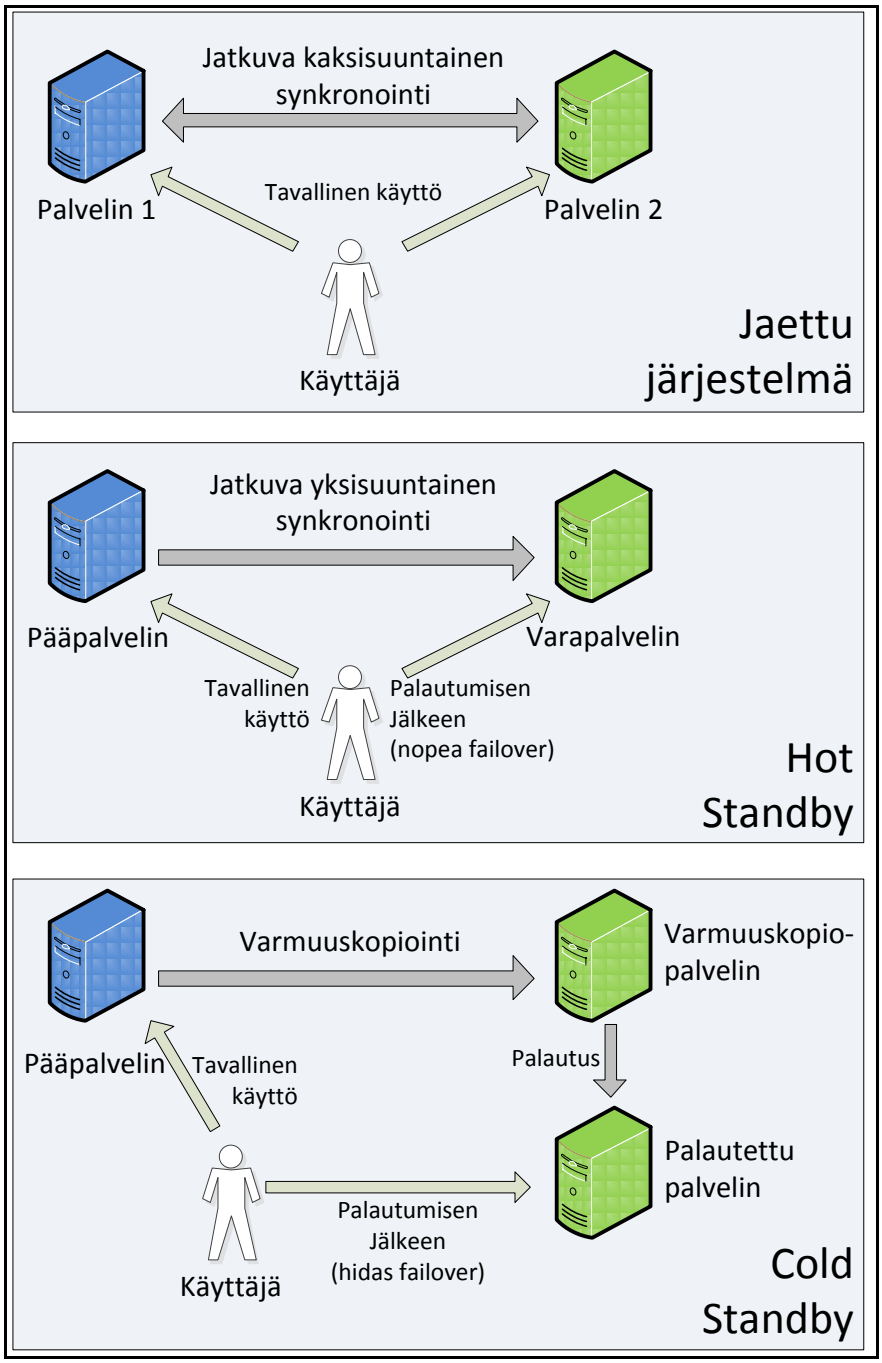
Käytännössä palautumisella tarkoitetaan prosessia katkoksen jälkeen, jolloin osa palvelun osista tai koko palvelu joudutaan asentamaan uudelleen, jotta se saataisiin taas

käyttöön. Useimmiten palautuminen suoritetaan toiselle palvelimelle, jonka jälkeen rikki mennyttä laitetta ryhdytään korjaamaan. Palautumiseen liittyy oleellisesti termi failover, jolla kuvataan palvelun siirtoa toiseen paikkaan käyttökatkoksen tapahtuessa. Tämä on usein automaattinen prosessi, mutta ei aina. SaaS-palveluissa se tarkoittaa yleensä siirtymistä varapalvelimeen, mikäli pääpalvelin kaatuu. [19.]

Palautumisprosessi riippuu palautusjärjestelmän arkkitehtuurista. Kaikkein kehittyneimmissä järjestelmissä heti kun palvelu menee alas, siirrytään automaattisesti käyttämään varajärjestelmää, eikä käyttäjä huomaa mitään eroa (Hot Standby). Toisessa ääripäässä ovat taas palautusjärjestelmät, joissa koko palvelu täytyy infrastruktuuria ja ohjelmistoa myöten asentaa käsin uudelle laitteelle. Useimmiten oikeissa tapauksissa käytetyt ratkaisut ovat jotain näiden väliltä. [5, s. 300–303.]

Tyypillisesti palautusjärjestelmät noudattavat yhtä kolmesta eri arkkitehtuurista. Nämä ovat jaettu järjestelmä, Hot Standby sekä Cold Standby (kuva 2). Jaetussa järjestelmässä kaikki palvelimet työskentelevät yhteistyössä ja liikenne palveluun on jaettu niiden kesken. Tämä on ylivoimaisesti vaikein järjestelmä toteuttaa, sillä se vaatii myös merkittäviä investointeja. Se on teoriassa myös toimintavarmin, sillä jos yksi palvelin kaatuu, ei palvelulle koidu minkäänlaista katkosta. Käytännössä ei aina näin ole, sillä molemmat palvelimet pitää saada toimimaan täydellisesti myös yksin. Hot Standbyssa tavallisesti käytetään vain yhtä palvelinta, mutta se jatkuvasti replikoi datansa varapalvelimelle. Palvelukatkoksen tullessa ohjataan käyttäjien liikenne automaattisesti varapalvelimelle, joka jatkaa siitä mihin pääpalvelin jäi. Cold Standby -järjestelmät toimivat muuten samalla tavalla, mutta palvelu pitää erikseen palauttaa, mikäli tarvetta sille on. Tämä on usein manuaalinen prosessi. [5, s. 300–302.]

Cold Standby -järjestelmät voidaan jakaa kahteen eri kategoriaan. Joissain tapauksissa kaikki tarvittava ohjelmisto on valmiina myös varapalvelimella, jolloin palautumiseen tarvitaan vain data. Joskus taas kaikki ohjelmistot ja konfiguraatiot täytyy asentaa palvelimelle erikseen. [5, s. 302.]



Kuva 2. Erilaiset palautusjärjestelmät [5, s. 301.]

Vaikka jaettu järjestelmä vaikuttaakin parhaalta vaihtoehdolta, ei näin aina ole, sillä kaksisuuntaista synkronointia varten tarvitaan aina komponentti, joka suorittaa tämän kuormantasauksen. Mikäli se menee rikki, hajoaa koko järjestelmä. Kuten kappaleessa 3 todettiin, ovat oikeat palautusjärjestelmät kompromisseja redundanssin ja yksinkertaisuuden välillä. Näin ollen usein ei käytetä puhtaasti mitään kolmesta esitellystä mallista, vaan palvelimilla voi olla muitakin tehtäviä kuin odottaa palvelukatkosta. [5, s. 302–303.]

5 Yrityksen SaaS-palvelut

Yrityksellä on yhteensä viisi web-sovellusta, jotka toimivat SaaS-periaatteella. Nämä ovat nimeltään TypingMaster Online, TypingMaster Online Single, TypingMaster LMS, AssessTyping sekä TypingTest [20]. Näistä viimeinen on ilmainen palvelu, jonka hyöty perustuu täysin sen mainostuottoihin sekä yrityksen tuotteiden promotointiin. Muut ovat pääasiassa maksullisia, mutta saattavat sisältää ilmaisen kokeilumahdollisuuden.

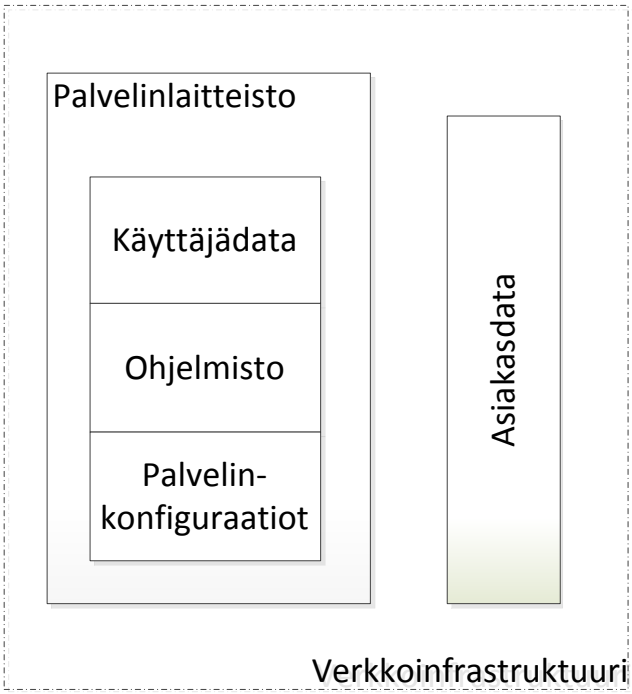
TypingMaster Online on web-sovellus, johon kirjaudutaan sisään tunnuksilla sekä uniikilla ID:llä. Se on kattava kymmensormijärjestelmän opetteluun tarkoitettu ohjelmisto, joka on suunnattu kouluille ja yrityksille [21]. Se on jaettu hallinta- ja opiskeluosioihin ja koostuu interaktiivisista kursseista, testeistä, peleistä ja tuloksien hallinnasta. TypingMaster Online Single on vastaava sovellus, mutta se on suunnattu yksityishenkilöille, joten siinä ei ole hallintaosiota. TypingMaster LMS taas on mahdollista integroida SCORM-pohjaiseen LMS:ään (Learning Management System).

AssessTyping on lähinnä yrityksille suunnattu kevyt webbisovellus, jonka avulla voidaan arvioida työntekijän, työnhakijan tai muun henkilön näppäilytaitoja [22]. TypingTest taas on ilmainen webbisivusto, jossa kuka vain voi monipuolisten testien avulla koetella taitojaan kymmensormijärjestelmä tai näppäilynopeudessa ja -tarkkuudessa [23]. Ilmaisuudesta huolimatta se on TypingMaster Onlinen ohella yksi business-kriittisimmistä palveluista.

Kaikki yrityksen palvelut noudattavat SaaS-ajatusmaailmaa. Asiakkaan näkökulmasta palvelun käyttöönotto on kaksivaiheinen: Ensin selaimella mennään verkkosivulle, jonka jälkeen kirjaudutaan sisään. Tämän jälkeen palvelu on täysin asiakkaan käytettävissä. Lähtökohtana on, että palveluiden täytyy olla saatavilla milloin tahansa ja mihin kellonaikaan tahansa ($24 \times 7 \times 365$). Erityisen tärkeää on datan eheys ja suojaus. Käytännössä tällä tarkoitetaan tässä tapauksessa SQL-relaatiotietokantoja. Nämä kaksi periaatetta luovat vahvan tarpeen palveluiden toimintavarmuudelle.

Koska suurin osa yrityksen myynnistä tapahtuu ulkomaille, varsinkin Yhdysvaltoihin, sijaitsevat kaikki SaaS-palveluja pyörittävät palvelimet USA:ssa.

Yrityksen kaikki SaaS-palvelut on rakennettu saman kaavan mukaan. Ne kaikki vaativat toimiakseen samankaltaisen palveluinfrastruktuurin (kuva 3). Peruspalikkana on yrityksen verkkoinfrastruktuuri, jonka päälle palvelut on rakennettu. Ne kaikki vaativat palvelinlaitteiston, joka taas sisältää kolme oleellista komponenttia: palvelinkonfiguraatiot, ohjelmiston sekä käyttäjädatan. Nämä ovat ne peruspalikat, joiden varassa kaikki yrityksen SaaS-palvelut toimivat. Samaan järjestelmään liittyy oleellisesti myös asiakasdata, joka sisältää muun muassa asiakas-, helpdesk- ja laskutustiedot. Ilman näitä olisi liikennetoiminnan harjoittaminen ongelmallista.



Kuva 3. Yrityksen SaaS-palveluiden infrastruktuuri

SaaS-palvelujen palvelinlaitteisto on ulkoistettu kolmannelle osapuolelle, joten vastuu sen toiminnasta ei ole yrityksen harteilla, kunhan vain prosessointitehoja on ostotilanteessa hankittu tarpeeksi. Kaikki palvelimen sisällä tapahtuva toiminta taas on yrityksen omaa aluetta, joten oleellista on juuri näiden osa-alueiden toimintavarmuuden maksimointi. Palvelimessa sijaitsevista komponenteista kaikki ovat oleellisia palvelun toimivuuden kannalta. Palvelinkonfiguraatiot koostuvat käyttöjärjestelmästä, erilaisesta Middlewaresta (väliohjelmisto, esim. ajurit) sekä palvelutoimintaa tukevista ohjelmistokonfiguraatioista. Käyttöjärjestelmänä yrityksen SaaS-palveluissa tällä hetkellä käytössä on Linux Ubuntu versio 10.04. Tähän kategoriaan kuuluvia konfiguraatioita ovat muun

muassa iptables-säännöt, lämpö- ja RAID- valvontaskriptit sekä FTP- ja SSH-palvelinohjelmistojen asetukset.

Kaikkien webbisovellusten ohjelmistototeutus on yksinkertainen ja ennen kaikkea yhtenevä. Ohjelmisto koostuu Apache Tomcat -palvelinohjelmistosta (käytössä versio 6.0), Java-ohjelmistoalustasta sekä MySQL-relaatiotietokantaohjelmistosta. Lisäksi välissä on joitain pienempiä komponentteja, kuten esimerkiksi Tomcatin vaatima MySQL-ajuri. Ohjelmistot pohjautuvat avoimen lähdekoodin ideologiaan, ja ne on suunniteltu toimimaan Linux-pohjaisilla palvelimilla, mutta tarpeen vaatiessa ne toimivat myös Windows-käyttöjärjestelmässä. Kaikki SaaS-palvelut pyörivät palvelimilla, joissa on Linux-käyttöjärjestelmä. Palvelut on eritelty Tomcatin Virtual Hosts -toiminnolla, joka mahdollistaa useiden eri domain-tason verkkosivujen ylläpidon samalla palvelimella.

Käyttäjädatalla tarkoitetaan yksinomaan palvelujen MySQL-tietokantoja. Kullakin palvelulla on oma tietokantansa, joiden taulurakenteet ovat yksilöllisiä. Palvelun jatkuvuuden kannalta tämä data on kaikkein oleellisinta, sillä se on sitä materiaalia, jota käyttäjät näkevät ja tuottavat. Sen täytyy siis aina olla eheää, reaaliaikaista, totuudellista sekä hyvin suojattua.

Liiketoiminnan jatkuvuuden kannalta oleellista on myös asiakasdata, joka koostuu käytännössä erilaisesta asiakashallintajärjestelmän datasta. Tätä ovat muun muassa laskutus- ja tilaustiedot sekä viestit. Vaikka asiakasdata ei vaikutakaan SaaS-palvelujen toimivuuteen, on se silti business-kriittinen komponentti, sillä sen avulla pidetään huoli siitä, kuka, miten ja milloin saa käyttää palveluja. Asiakasdata ei sijaitse SaaS-palvelimilla vaan yrityksen omassa Intranetissä.

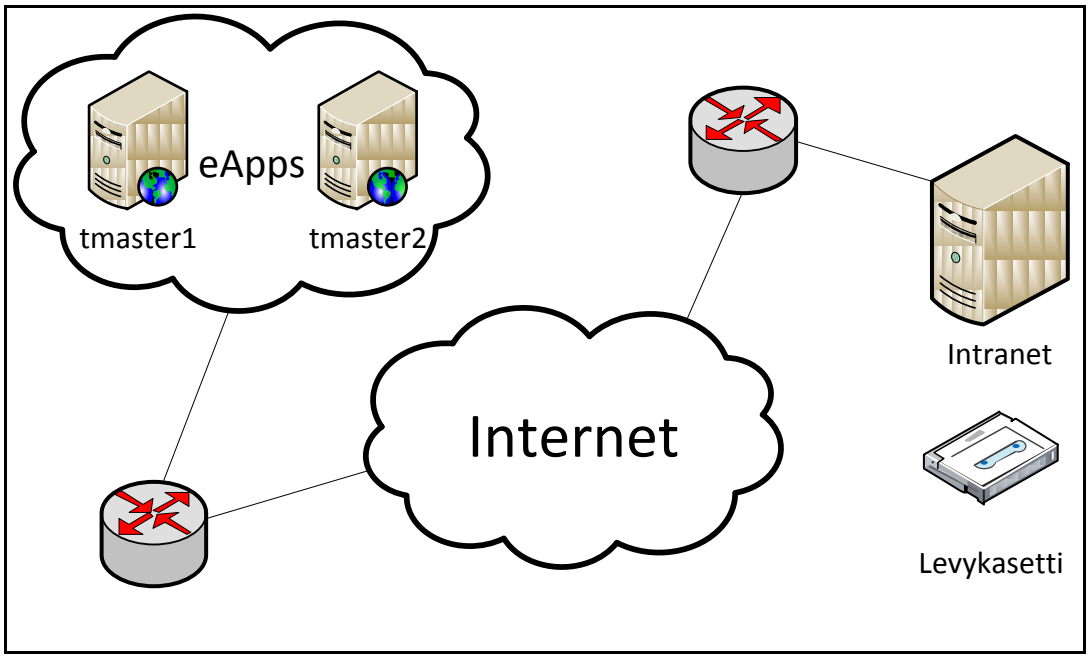
Yhdessä nämä palikat muodostavat palveluinfrastruktuurin, jonka varassa yrityksen SaaS-palvelut toimivat. Ne kaikki ovat oleellisia palvelujen toimintavarmuuden kannalta, mutta erityistä huomiota vaatii käyttäjädata. Palvelinkonfiguraatiot ja ohjelmistot ovat hyvin pitkälti staattisia komponentteja, eikä niille tarvitse kovinkaan usein tehdä mitään. Sitä vastoin käyttäjädata muuttuu ja elää koko ajan, joten siitä täytyy pitää erityistä huolta [24].

Yrityksen palveluiden toimintavarmuuteen vaikuttavat myös muutamat muut tekijät, kuten esimerkiksi ulkoistetut DNS- ja sähköpostipalvelut. Näiden lisäksi myös virheentunnistuksen ja ilmoitusten täytyy toimia, jotta palvelun käyttökatkokset tai pienemmät virheet havaitaan ajoissa.

6 Vanha tilanne

6.1 Verkkoinfrastruktuuri

Aikaisemmin yrityksen verkkoinfrastruktuuri oli hyvin yksinkertainen ja jakaantui kahteen alueeseen: intranettiin (tukiverkko) ja ulkoistettuihin web-palvelimiin (palveluverkko). Intranet oli tässä vaiheessa hyvin pitkälti yksinomaan yrityksen sisäisessä käytössä, eikä sen rooli web-palveluja tukevana elementtinä ollut kovinkaan vahva. Palvelut pyörivät itsenäisesti eikä automatisointia tai redundanssia juurikaan ollut. Kriittisten virheiden tunnistus toimi, mutta sähköpostivaroitus tuli lähinnä vain, jos koko palvelin oli alhaalla. Mikäli jotakin meni pieleen, täytyi palautuminen tehdä lähes täysin manuaalisesti, eikä täydellinen palautuminen ollut aina edes mahdollista.



Kuva 4. Verkkoinfrastruktuurin vanha tilanne

Vuonna 2010 tapahtuikin tilanne, jolloin eApps-palveluntarjoajalla sijaitsevan virtuaalipalvelimen RAID-levypakka hajosi. Tämän myötä kaikki TM Online -palvelun data hävisi. Palvelun varmennus oli vielä tällöin kahden tekniikan varassa, jotka olivat RAID-varmennus sekä päivittäiset tietokantavarmuuskopiot. Koska itse RAID-varmennus hajosi, ei jäljelle jäänyt muuta kuin lähes vuorokausi aikaisemmin otettu varmuuskopio. Kun uusi palvelin saatiin jälleen pystyyn, palautettiin siihen tietokannat tästä varmuuskopiosta. Vahinko oli kuitenkin jo tapahtunut, ja viimeisen vuorokauden ajalta kaikki käyttäjädata hävisi. Myöhemmin palveluntarjoaja onnistui palauttamaan jonkin verran sekalaista dataa rikki menneeltä kiintolevyltä, mutta eheään palautumiseen ei koskaan päästy.

Varsinainen laitteistoinfrastruktuuri oli palveluntarjoajien vastuulla, joita velvoitti sopimuksen mukainen SLA. Tästä huolimatta verkko- ja laitteisto-ongelmia ilmeni varsinkin eApps-palveluntarjoajan kanssa. Sisäverkon infrastruktuuri oli luonnollisesti yrityksen vastuulla, lukuun ottamatta Internet-yhteyttä.

6.1.1 Palveluverkko

Palveluverkko eli tuotantoverkko koostui kahdesta useita vuosia käytössä olleesta virtuaalipalvelimesta (VPS, Virtual Private Server), joissa toimi hajautettuna viisi webbisovellusta (TM Online, TM Online Single, TM LMS, AssessTyping, TypingTest). Virtuaalipalvelimet oli ostettu palveluntarjoajalta nimeltä eApps, jonka palvelinsali sijaitsi fyysisesti Atlantassa, Yhdysvalloissa. Molempiin oli osoitettu lähes kiinteä määrä suoritustehoja ja muistia, ja verkkoliikenteen määrä oli rajoitettu. Rajat olivat kuitenkin sellaisia, ettei niiden ylittäminen ollut todennäköistä. Laitteisto- ja verkkoinfrastruktuurin hoito oli sopimuksen mukaisesti palveluntarjoajan vastuulla. Tämän piti säästää yrityksen IT-henkilöstöltä vaadittavaa työpanosta. Pilvipalvelun (IaaS) mukaisesti periaatteena oli, että yritys saa vapaasti käyttää vuokraamiaan virtuaalipalvelimia kuten haluaa, eikä ympäröivästä infrastruktuurista tarvitse huolehtia.

Kaikesta huolimatta muutaman vuoden käytön jälkeen varoituksia resurssiylityksestä alkoi tulla, eikä palveluohjelmien ajo aina toiminut kuten piti. Eniten ongelmia vuoden 2011 lopulla aiheuttivat Tomcat- ja Java-sovellukset, sillä palveluntarjoajan mukaan niiden muistinkäyttö kohosi välillä yli sallitun rajan. Syy tähän jäi epäselväksi, sillä

Tomcat-instanssille oli allokoitu kiinteä määrä resursseja. Erityistä päänvaivaa aiheutti myös virtuaalipalvelimen kiintolevy, joka fyysisesti oli jaettu muiden käyttäjien kanssa. Vaikka kiintolevyn kapasiteetti olikin allokoitu, ei sen kirjoitustehoa ja -nopeutta käytännössä ollut mahdollista jakaa tasapuolisesti käyttäjien kesken. Tämä aiheutti sen, että muiden virtuaalipalvelinasiakkaiden prosessit käyttivät suuren määrän levyn kirjoitustehoa. Tämä taas johti esimerkiksi MySQLDump-varmuuskopioinnin jumittumiseen, sillä se oli konfiguroitu toimimaan vain silloin, kun mikään muu prosessi ei käytä levyä. Varsinainen syy siihen, miksi ongelmia alkoi ilmetä vasta vuonna 2011, on epäselvä. Tilanne vaikutti kumminkin vahvasti siltä, että palveluntarjoaja ajoi aivan liian montaa virtuaalikonetta samalla palvelimella.

Vuokratuissa virtuaalipalvelimissa ei ollut nykystandardien mukaan paljoa tehoja, mutta yrityksen palveluiden pyörittämiseen niiden olisi pitänyt riittää (taulukko 1).

Taulukko 1. Vanhojen eApps VPS-palvelimien spesifikaatiot

Muisti	Toisessa 700 MB ja toisessa 2 GB
Kiintolevy	Toisessa 2 x 10 GB + RAID Controller Toisessa 2 x 30 GB + RAID Controller

6.1.2 Tukiverkko

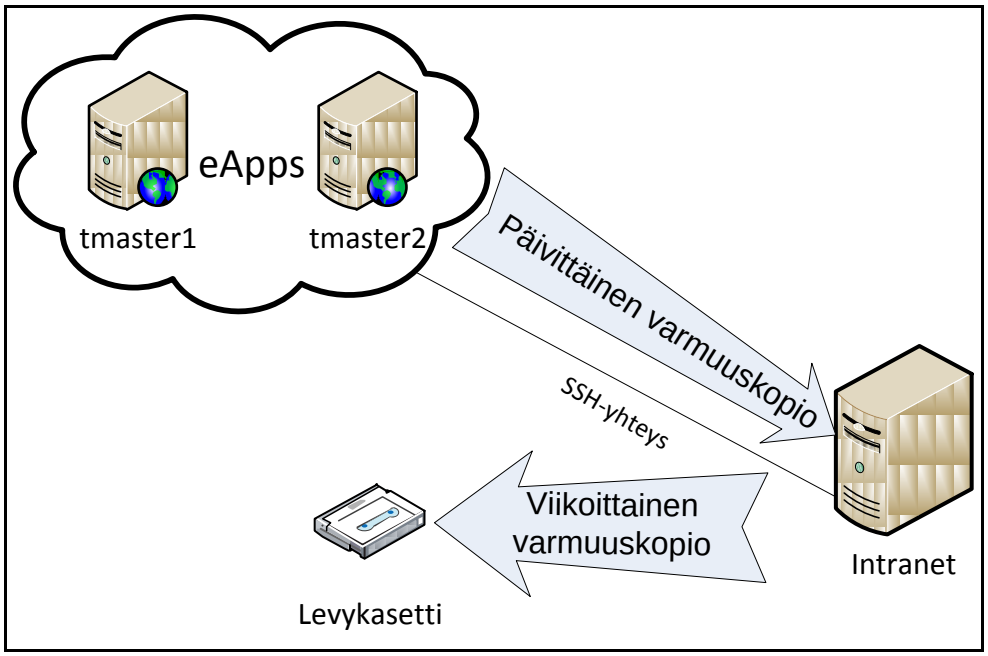
Yrityksen tukiverkko koostui kahdesta Intranetissä sijaitsevasta palvelimesta, joista vain toinen oli tekemisissä SaaS-palvelujen kanssa, sekä ulkoistetuista DNS- ja sähköpostipalveluista. Molemmissa Intranet-palvelimissa oli Windows Server 2003 -käyttöjärjestelmä. Toinen palvelimista oli lähinnä tuotekehityskäytössä, ja toisessa pyörivät erilaiset oleelliset sisäverkon palvelut. Lisäksi siihen ladattiin yrityksen verkkopalvelujen päivittäiset varmuuskopiot. Intranet-palvelimen rooli SaaS-palveluja tukevana elementtinä oli siis hyvin yksinkertainen, sillä se toimi vain varastona päivittäisille varmuuskopioille. Lisäksi kyseisestä palvelimesta otettiin viikoittain varmuuskopio levykasetille.

6.2 Toimintavarmuus

SaaS-palvelujen toimintavarmuutta ei ollut tässä vaiheessa vielä mietitty kovinkaan pitkälle. Tärkeintä oli, että palvelut pyörivät. Mikäli häiriöitä ilmeni, pyrittiin niihin puuttumaan manuaalisesti mahdollisimman pian. IaaS-periaatteiden mukaisesti palvelimien ja verkon toimintavarmuus oli eApps-palveluntarjoajan vastuulla. Vikasietoisuutta tai onnettomuuksiin varautumista ei tällöin vielä juurikaan ollut. Mikäli onnettomuus tapahtui, reagoitiin siihen tapauskohtaisesti.

6.2.1 Varmennus ja varmuuskopiointi

Varmuuskopiointi pyöri pitkälti Intranet-palvelimen ympärillä (kuva 5). Siinä pyöri Windows Task Schedulerilla, Batch-skripteillä sekä WinSCP-ohjelmalla toteutettu järjestelmä, joka latasi palvelimelle päivittäin kaikkien palvelujen MySQL-varmuuskopiot. Varmuuskopiot oli toteutettu MySQL:n omalla MySQLDump-ohjelmalla, joka automaattisesti loi dumpin halutuista tietokannoista joka päivä. Se suoritettiin päivittäin Cron-tehtävänä. Tämän jälkeen Intranet-palvelimella päivittäin pyörivä Batch-skripti otti WinSCP-ohjelman avulla SSH-yhteyden sekä tmaster1- että tmaster2-palvelimiin ja latasi molemmista MySQL dumpit talteen. Näitä pidettiin palvelimella kahden viikon ajan.



Kuva 5. Varmuuskopiointijärjestelmän vanha tilanne

Intranet-palvelimesta otettiin viikoittain myös täydellinen varmuuskopio levykasetille, johon sisältyi kaikkien palveluiden varmuuskopio. Tämä oli varmuuskopioitavan datan luonteeseen nähden todella raskas ja vaivalloinen prosessi, sillä kasetti piti vaihtaa käsin.

MySQL dumpin luominen eApps-palveluntarjoajan virtuaalipalvelimilla loi pieniä ongelmia palvelimien suoritustehojen kanssa, sillä se tehtiin suoraan aktiivisesta tietokannasta. Parhaillaan tätä samaa tietokantaa saattoivat käyttää sadat eri käyttäjät. Ongelmaa lievitti Nice-työkalu, jolla annettiin MySQLDump-ohjelmalle pienin mahdollinen prioriteetti, jonka ansiosta dumppia tehtiin vain silloin, kun suoritustehoja ja muistia oli vapaana. Myöhemmin tämä tosin johti palveluntarjoajan ongelmien takia dumpin muodostuksen jumittumiseen.

Varmuuskopiointia tehtiin myös tuki-infrastruktuurin tasolla. Verkkolevyistä ja yrityksen sisäisistä palveluista otettiin aiemmin varmuuskopio viikoittain levykasetille. Lisäksi päivittäiset inkrementaaliset varmuuskopiot otettiin toiselle sisäisessä käytössä olevalle palvelimelle, mutta se poistui käytöstä vuoden 2011 alussa.

6.2.2 Virheentunnistus ja ilmoitukset

Jotta palvelun toiminnasta voitaisiin olla varmoja, täytyi sitä monitoroida. Ohjelmisto- ja käyttäjävirheet ovat asia erikseen, joten niitä ei tässä työssä käsitellä. Infrastruktuuriin ja palvelun toimintaan tai toimimattomuuteen liittyvät virheet taas ovat erittäin olennainen asia. Jotta virheet ja käyttökatkokset havaittaisiin ajoissa, täytyy niistä tulla ilmoitus. Tämä oli toteutettu palveluntarjoajan omilla virheviesteillä sekä SiteUptime-palvelulla.

Palveluntarjoaja eApps lähetti automaattisesti sähköpostiviestejä, mikäli palvelimella tapahtui kriittinen virhe tai jos käytetyt prosessointitehot ylittivät myönnetyn rajan. SiteUptime taas on palvelu, joka tietyin väliajoin (5 minuutin välein) tarkistaa palvelun saatavuuden. Mikäli palveluun ei saada yhteyttä, lähettää se virheviestin. Tämä oli asetettu monitoroimaan kaikkia yrityksen palveluja. Varmuuskopiointi- ja varmennusvirheistä ei tällöin vielä tullut ilmoituksia.

7 Muutokset ja nykytilanne

7.1 Tavoitteet

Uudistuksiin SaaS-palvelujen suhteen päätettiin ryhtyä jo vuoden 2011 alkupuolella, ja niitä toteutettiin hiljalleen kesällä ja syksyllä. Viimeiset uudistukset tehtiin vielä 2012 alussa. Syy uudistusten tarpeeseen oli ilmiselvä, sillä uskottavaa SaaS-palvelua on todella vaikeaa (tai vähintäänkin kyseenalaista) myydä ja ylläpitää ilman kunnollista toimintavarmuutta. Vahva tarve toimintavarmuuden parantamiselle siis oli, mutta se, miten tämä toteutettaisiin, piti vielä suunnitella. Avainsanoja kehitykselle olivat tehokkuus, yksinkertaisuus ja turvallisuus. Uudistukset tehtiin näiden pohjalta.

Toiminnallisuuksien osalta yrityksen SaaS-palvelut pysyivät pääpiirteittäin samana. Sisältöä ja uusia ominaisuuksia niihin tuli, mutta tämä ei vaikuttanut mitenkään ohjelmistoihin tai infrastruktuuriin, jossa ne toimivat. Myös ohjelmistoinfrastruktuuri pysyi hyvin pitkälti samana. Pääpalikat olivat edelleen Linux-käyttöjärjestelmä, Java, Apache Tomcat sekä MySQL.

Palveluiden verkkoinfrastruktuuri päätettiin muuttaa. Edellisen IaaS-palveluntarjoajan ilmettyä epäluotettavaksi, päätettiin siirtyä toisen palveluntarjoajan Managed Hosting-ratkaisuun. Tällöin jaetun virtuaalipalvelimen sijaan palvelinraudan omistus ja hallinta ulkoistettiin Managed Hosting -palveluntarjoajalla. Yritys sai siis käyttöönsä täysin itselleen dedikoidun palvelinlaitteiston, eikä sen suoritustehoja tarvitse enää jakaa muiden kanssa. Vaikka tämä onkin hiukan kalliimpi ratkaisu kuin IaaS-periaatteella toimiva virtuaalipalvelin, on se myös merkittävästi luotettavampi. Vastuu palvelimen laitteisto- ja verkkoinfrastruktuurin toimivuudesta on edelleen palveluntarjoajalla. [2, s. 68–69.]

Merkittäviä muutoksia päätettiin tehdä myös yrityksen sisäverkkoon. Osa muutoksista vaikutti SaaS-palveluihin tukiverkon muodossa ja osa taas ei. Merkittävin muutos oli Extranet-palvelimen hankinta. Sen suurin rooli on yrityksen sisäisessä käytössä, mutta lisäksi se otti myös Intranet-palvelimen roolin varmuuskopioiden sijoituspaikkana. Intranet-palvelinkaan ei tosin jäänyt käyttämättä.

Muutoksilla pyrittiin varmennettuun SaaS-järjestelmään, joka mahdollistaisi kiitettävän toimintavarmuuden. Järjestelmää suunnitellessa täytyi ottaa huomioon monia seikkoja. Vikasietoisuus ja redundanssi olivat erityisen tärkeitä avainsanoja. Mikäli toinen SaaS-palvelimista kaatuu, täytyy toisen pystyä väliaikaisesti hoitamaan sen roolia. Järjestelmän tuli silti olla yksinkertainen, sillä liian moni liikkuva osa vain monimutkaistaisi asiaa. Myös kompaktiuteen ja taloudellisuuteen pyrittiin, eikä esimerkiksi pääasiassa varalle jääviä palvelimia ollut järkevää hankkia. Maksullisten ratkaisujen sijaan käytettiin avoimeen lähdekoodiin perustuvia ohjelmia, sillä niiden arvioitiin olevan tarpeeksi hyviä. Jotta haluttu järjestelmä saataisiin toteutettua, tuli muutoksia tehdä ennen kaikkea verkkoinfrastruktuuriin. Kun tämä oli valmis, voitaisiin varmuuskopiojärjestelmä rakentaa sen päälle.

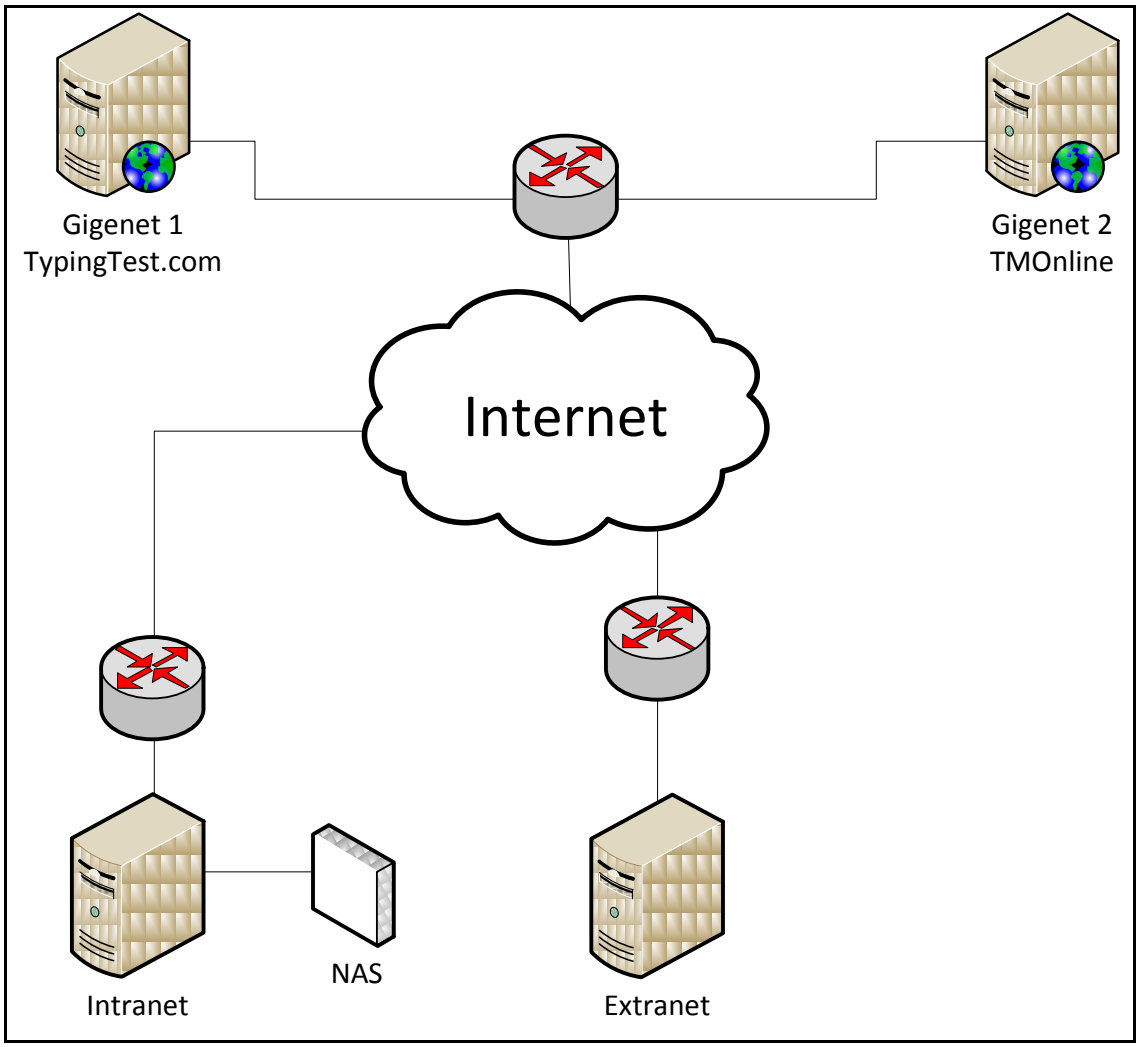
Uusi Extranet-palvelin kahden uuden vuokrapalvelimen ohella muodosti kolmion, jonka ympärille palveluiden varmennus- ja varmuuskopiojärjestelmä pyrittiin rakentamaan. Kahden erillisen SaaS-palvelimen ansiosta päätettiin rakentaa ristikkäinen Cold Standby -järjestelmä, jonka ansiosta varmuuskopiointi ei häiritse käytössä olevia palveluita.

Käyttäjädatan varmuuskopioinnin lisäksi päätettiin myös yrityksen sisäisiä palveluja varten rakentaa varmuuskopiointijärjestelmä. Vaikka tähän kuuluukin dataa, joka ei liity SaaS-palveluihin, on siinä mukana myös palveluiden ylläpidon, palautumisen ja jatkuvuuden kannalta tärkeää materiaalia. Osa niistä vaikuttaa palveluihin suoraan, kuten esimerkiksi palvelinkonfiguraatiot, ja osa taas epäsuorasti, kuten esimerkiksi asiakasdata.

7.2 Verkkoinfrastruktuuri

Yrityksen verkkoinfrastruktuuria päivittyi ja muuttui paljon. Interaktiivisuus SaaS-verkon ja tukiverkon välillä oli ennen ollut erittäin yksinkertaista ja yksisuuntaista. Varmuuskopiot vain ladattiin paikasta A paikkaan B. Tätä järjestelmää haluttiin laajentaa, joten toimintavarmuuden lisäämiseksi päätettiin koko verkkoinfrastruktuuria uusia. Syynä tähän oli ilmeinen SaaS-palvelujen suosion kasvu sekä liiketoiminnan kannalta kriittinen toimintavarmuus. Uudistukset eivät koskeneet pelkästään palveluverkkoa, vaan myös tukiverkko uudistui. Osa näistä uudistuksista koski yrityksen sisäistä verk-

koa, ja osa vaikutti myös SaaS-palveluihin. Suurimmat muutokset verkkoinfrastruktuurissa olivat eApps-palveluntarjoajan vaihto Gigenet-palveluntarjoajaan, sekä Extranet-palvelimen hankinta (kuva 6).



Kuva 6. Verkkoinfrastruktuurin uusi tilanne

Tavoitteena oli, että verkkoinfrastruktuurista tulisi yksinkertainen, mutta äärimmäisen tehokas ja toimintavarma. Ennen kaikkea infrastruktuurin täytyi olla sellainen, että siinä pystytään tehokkaasti pyörittämään uutta varmennus- ja varmuuskopiojärjestelmää.

7.2.1 Palveluverkko

Yrityksen SaaS-palvelut siirrettiin uudelle palveluntarjoajalle nimeltä Gigenet, jonka palvelinsali sijaitsee Chicagossa, Yhdysvalloissa. Virtuaalipalvelimien sijaan käyttöön

otettiin kaksi laitteistoltaan identtistä dedikoitua palvelinta (taulukko 2), jotka ovat fyysisesti erillään, mutta sijaitsevat kumminkin samassa palvelinsalissa. Palvelinten fyysinen erillisyys on välttämätöntä halutun vikasietoisuuden kannalta, sillä tällöin pitäisi koko palvelinsalille tapahtua katastrofi, jotta palveluille aiheutuisi huomattavaa haittaa. Lisäksi fyysisen laitteiston vuokraus mahdollistaa itse toteutetun virtualisoinnin, mikäli tarvetta tälle ilmenee.

Taulukko 2. Uusien Gigenet-palvelimien spesifikaatiot

Prosessori	2 x AMD Opteron 2212, 2 GHz
Muisti	2 GB
Kiintolevy	2 x 80 GB + RAID Controller
Käyttöjärjestelmä	Ubuntu 10.04 (Natty)

Palveluiden siirto uusille palvelimille tapahtui palvelu kerrallaan, jotta niiden toimivuus ehdittiin testata huolellisesti ennen seuraavan siirtoa. Palveluiden jako palvelimiin tapahtui niin, että toisesta palvelimesta tehtiin TypingTest-palvelin massiivisen kävijämäärän takia, ja toiseen taas sijoitettiin yrityksen maksulliset Online-palvelut (kuva 6).

Jokaisen palvelun siirto tapahtui niin, että ensin vanhalla palvelimella pysäytettiin Tomcat-ohjelmisto, jonka jälkeen kunkin palvelun MySQL-tietokannasta otettiin dump, joka siirrettiin uudelle palvelimelle. Tässä vaiheessa uusille palvelimille oli jo asennettu tarvittavat ohjelmistot, joiden asennusvaiheet dokumentoitiin yrityksen sisäiseen käyttöön mahdollista vikatilannetta varten. Tämän jälkeen palveluiden MySQL-dumpit importoitiin uudelle palvelimelle ja Tomcat käynnistettiin. Kun kaikki toimi halutulla tavalla, vaihdettiin verkkosivun DNS-tietueet vastaamaan uuden palvelimen IP-osoitetta. Tällä tavalla data eli tietokannat saatiin pidettyä yhtenäisinä, eikä esimerkiksi päällekkäisyyttä tullut. Toimenpide suoritettiin jokaiselle palvelulle erikseen.

Siirron jälkeen palvelut toimivat uusilla palvelimilla halutulla tavalla. Joitain pieniä ohjelmistobugeja ilmeni, mutta ne liittyivät pääasiassa itse tuotteeseen ja korjattiin hyvin pian.

7.2.2 Tukiverkko

SaaS-palvelujen tukiverkko laajeni käsittämään vasta hankitun Extranet-palvelimen. Se vuokrattiin Managed Hosting -periaatteella Englannista, ja tuli pääasiassa yrityksen sisäiseen käyttöön. Extranet-palvelimella pyörii Windows 2008 R2 -käyttöjärjestelmä. Se otti Intranet-palvelimen roolin yritysten sisäisten palveluiden keskuksena sekä varmuuskopioiden varastona. Verkkolevyt jäivät edelleen Intranet-palvelimelle.

Uusi palvelin loi mahdollisuuden entistä monipuolisemman varmuuskopiointijärjestelmän luomiseen. Sen ulkoistamisen myötä vastuu palvelimen infrastruktuurin toiminnasta siirtyi palveluntarjoajalle ja lisäksi saatiin kaikille varmuuskopioille vielä yksi erillinen sijoituspaikka siltä varalta, jos vaikka yrityksen toimitiloille ja Intranet-palvelimelle sattuisi jotain. Erityisen suotuisaa varmuuskopioiden kannalta oli Extranet-palvelimen Internet-yhteyden nopeus, joka on 100 Mb/s molempiin suuntiin. Intranet-palvelimen yhteys ei pystynyt edes kymmenesosaan tästä.

Koska Intranet-palvelimen kiintolevyillä alkoi jo olla ikää noin kymmenen vuotta, päätettiin sen rinnalle hankkia NAS-laite (Network-attached Storage) eli verkkolevy. Tämä liitettiin Intranet-palvelimeen iSCSI-levynä. Vaikka operaatio liittyikin lähinnä yrityksen liiketoiminnan turvaamiseen, tuli uudesta verkkolevystä myös osa uutta varmuuskopiointijärjestelmää, jossa säilytetään muun muassa palveluiden kuukausittaiset varmuuskopiot.

Keskitettyä DNS:än hallintaa varten otettiin käyttöön uusi palveluntarjoaja nimeltä DNSMadeEasy, jonka avulla saatiin kaikkien palveluiden DNS-merkinnät saman palvelun alle. Palvelu tukee myös DNS failoveria, joka mahdollistaa palvelun monitoroinnin ja automaattisen vaihdon toiseen osoitteeseen. Tätä ei kuitenkaan vielä implementoitu uuteen järjestelmään.

7.3 Toimintavarmuus

Verkkoinfrastruktuurin uudistuksien myötä yrityksen palveluiden toimintavarmuus päätettiin saattaa uudelle tasolle. Apuna käytettiin kappaleessa 3 määriteltyjä periaatteita.

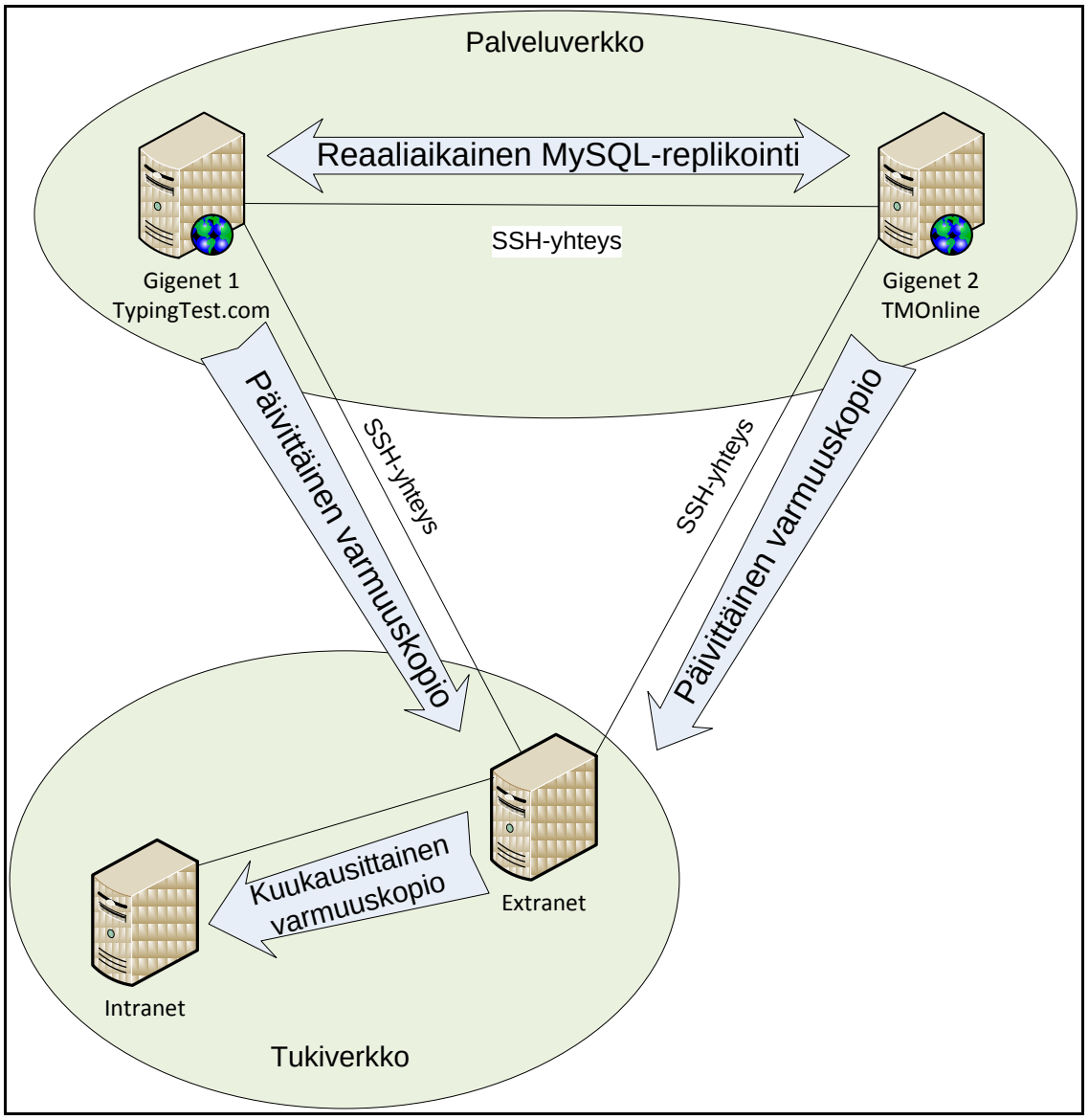
Tärkein painopiste yrityksen SaaS-palveluiden toimintavarmuudessa on datan varmennuksella ja varmuuskopioinnilla. Näitä molempia sovellettiin uudessa toimintavarmuusmallissa, jotta järjestelmän vikasietoisuus saataisiin maksimoitua.

Koska lähes kaikki yrityksen palvelut ja järjestelmät on rakennettu itse käsin, päätettiin myös varmennus- ja varmuuskopiointijärjestelmä rakentaa näin. Kuten luvussa 3.2.2 todettiin, erilaisia kaupallisia ratkaisuja on olemassa paljon. Tästä huolimatta Linux-pohjaisiin käyttöjärjestelmiin on olemassa niin hyviä avoimeen lähdekoodin perustuvia ohjelmistoratkaisuja, että järjestelmän rakentamiseen päätettiin käyttää yksinomaan näitä.

7.3.1 Palvelujen varmennus ja varmuuskopiointi

Yrityksen SaaS-palvelut päätettiin rakentaa järjestelmän varaan, joka koostuu sekä datan varmennuksesta että varmuuskopioinnista. Suurimman haasteen järjestelmän rakentamiselle loi palvelujen globaali suosio. Tämän ansiosta jokainen palvelu on käytössä ympäri vuorokauden, joten ideaalista kellonaikaa varmuuskopioinnille ei ole. Palveluiden hajauttaminen kahdelle eri palvelimelle tosin tarjosi tähän ratkaisun. Päätettiin rakentaa kolmesta palvelimesta koostuva järjestelmä (kuva 7), johon sovellettiin sekä Cold Standby että Hot Standby -ajattelua. Kaikki käyttäjädata viimeisen vuorokauden ajalta on tallessa useassa paikassa, ja reaaliaikainen data aina kahdessa paikassa. Lisäksi datasta otetaan kuukausittainen kopio vielä neljännelle palvelimelle.

Yrityksen palvelut on rakennettu niin, että varsinainen ohjelmistoinfrastruktuuri saadaan toimimaan hyvin pienellä vaivalla. Se on myös hyvin pitkälti staattinen kokonaisuus, sillä ohjelmien konfiguraatiot eivät pääsääntöisesti muutu. Tämän takia sitä ei sisällytetä varmuuskopioihin. Yrityksen omalla verkkolevyllä pidetään tosin valmiina käytössä olevien ohjelmien konfiguraatiot, mitkä on tarpeen tullen helppo siirtää palautetulle palvelimelle. Tästä kerrotaan enemmän luvussa 9.

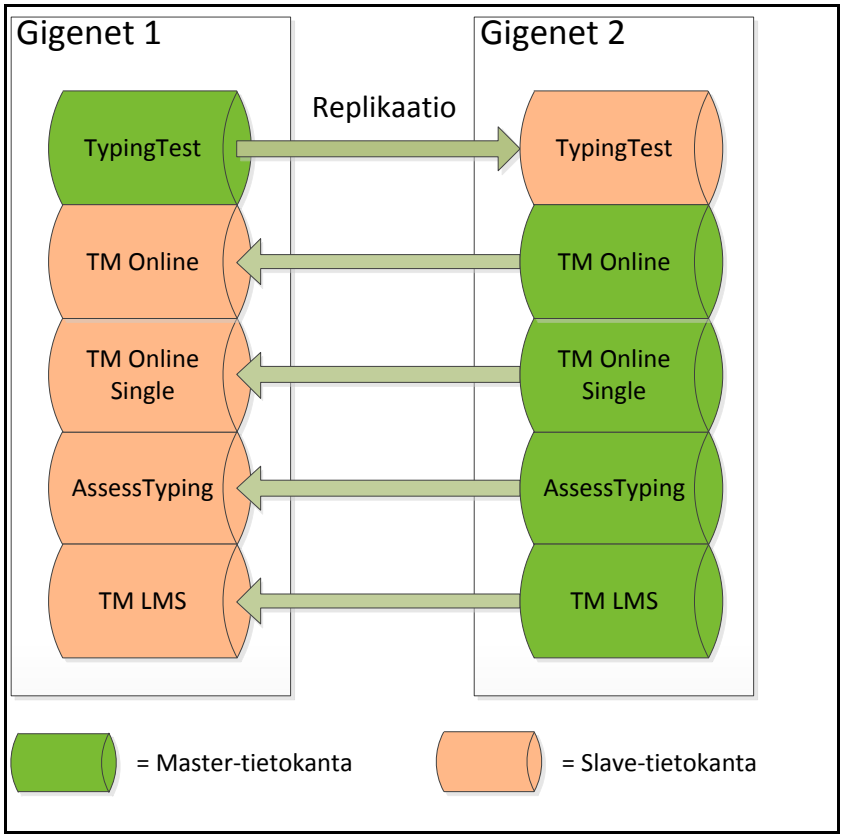


Kuva 7. Varmuuskopiointijärjestelmän uusi tilanne

Molemmissa uusissa Gigenet-palvelimissa on 3ware-valmistajan RAID Controller, jonka avulla palvelimien kiintolevyt on alustettu RAID 1 -levypakaksi. Kummatkin palvelimet sisältävät siis kaksi kiintolevyä, jotka on peilattu keskenään eli ne ovat sisällöltään identtiset. Vaikka tämä tarjoaakin ylimääräistä vikasietoisuutta, ei mahdollista palautumistilannetta jätetty RAID-levyn varaan, sillä se ei olisi kovinkaan tehokas toimintatapa.

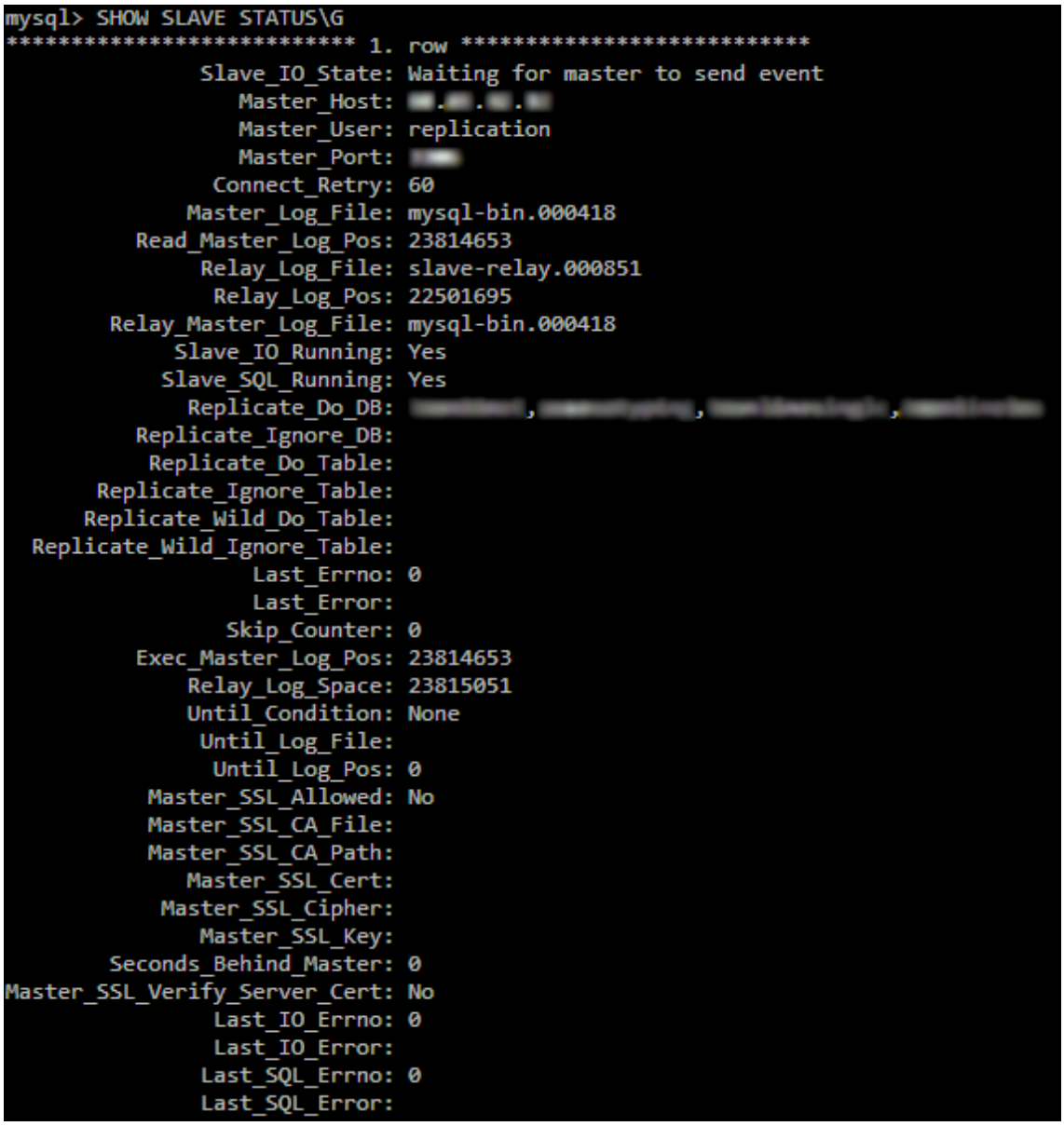
SaaS-palvelimiin päätettiin asentaa ristikkäinen MySQL-replikointi eli peilaus. Tämä tarkoittaa sitä, että kaikkien palveluiden tietokannat sijaitsevat molemmissa palvelimissa, mutta vain toiseen kirjoitetaan. Käytännössä tämä tehtiin niin, että jokainen tietokanta asetettiin pääpalvelimellaan Master-tilaan muokkaamalla my.cnf-tiedostoa, eli

MySQL:ää käskettiin kirjoittamaan erillistä binäärilokia jokaisesta tapahtumasta. Tämän jälkeen tietokantojen dumpit siirrettiin varapalvelimelle, johon ne myös asennettiin. Nämä tietokannat taas asetettiin my.cnf-tiedostoa muokkaamalla Slave-tilaan, minkä jälkeen MySQL:lle kerrottiin Master-palvelimen tiedot sekä binääriloki, jota sen tulisi lukea, ja Slave-toiminto käynnistettiin. Samat asennustoimet tehtiin ristikkäin molempiin palvelimiin, jonka jälkeen molemmilla palvelimilla oli kaikkien palveluiden tietokannat (kuva 8).



Kuva 8. SaaS-palvelimien MySQL-replikointi

Vaikka molemmilla palvelimilla sijaitseekin kaikkien palveluiden tietokannat, ei niitä valitettavasti voi käyttää täysin synkronoidun jaetun järjestelmän luomiseen, sillä MySQL ei virallisesti tue Multi-Master -replikointia [25]. Tällöin kirjoitettavat binäärilokit menisivät nopeasti ristiin, eikä järjestelmä toimisi. Tämän takia Master- ja Slave-tietokannat pidetään selvästi erillään.



Kuva 9. Toimivan Slave-tietokannan tila

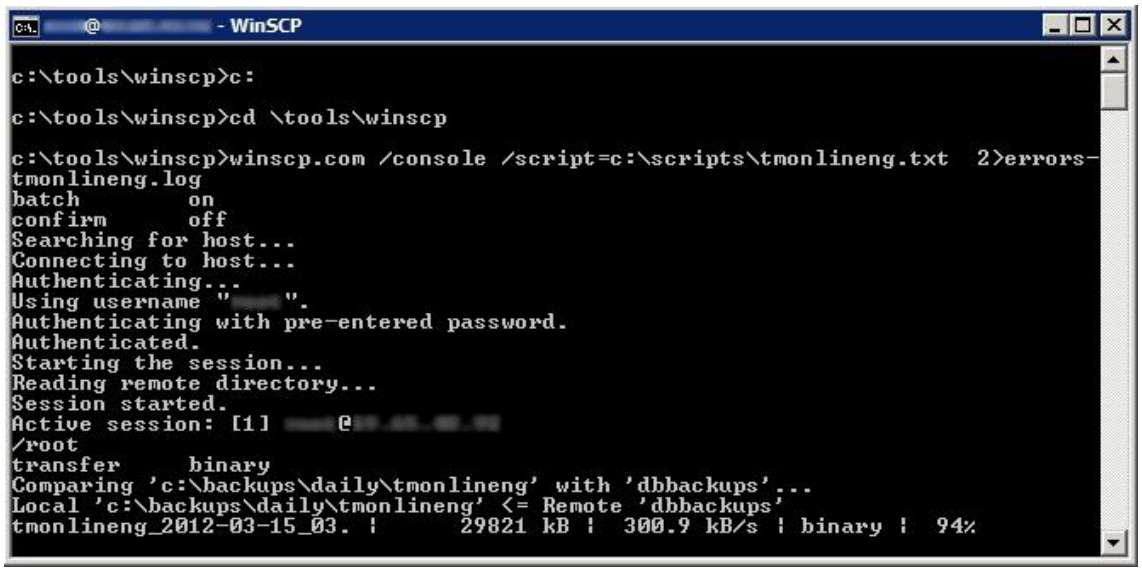
MySQL-replikaation avulla vikasietoisuutta saatiin tuotua Cold Standby -järjestelmästä lähemmäksi Hot Standbyta. Data sijaitsee molemmilla palvelimilla, mutta onnettomuuden sattuessa Slaven saattaminen Masteriksi vaatii edelleen muutaman yksinkertaisen toimenpiteen. Tällä tavoin myöskään käyttäjädataa ei pitäisi hävitä, sillä jos jotakin vakavaa pääpalvelimelle sattuu, on Slave-tietokanta jo suurella todennäköisyydellä ehtinyt kopioida itselleen viimeisimmät muutokset.

Koska molemmat Gigenet-palvelimet sijaitsevat saman palveluntarjoajan tiloissa, päätettiin vikasietoisuutta lisätä entisestään ottamalla tietokannoista päivittäiset varmuus-

kopiot. Tällöin kuvaan tuli mukaan uusi Extranet-palvelin. Molemmat SaaS-palvelimet asetettiin luomaan MySQL dump jokaisesta Slave-tietokannastaan joka päivä. Tämä toteutettiin asettamalla Cron ajamaan uusi DBBackup Bash-skripti (liite 1) jokaiselle palvelimen Slave-tietokannalle kerran päivässä. Käytännössä skripti poistaa yli 30 päivää vanhat dumpit, ajaa MySQLDump-ohjelman tietokannalle, pakkaa tehdyn dumpin gz-tiedostoksi sekä tarkistaa sen oikeellisuuden (tarkemmin kappaleessa 9.3.2). Ajot synkronoitiin niin, ettei niitä varmasti mene päällekkäin. Lisäksi käytettiin Nice-ohjelmistoa, jonka avulla MySQLDump-prosessille myönnettiin prosessointitehoja vain silloin, kun niitä on vapaana.

Koska varmuuskopiot (dumpit) otetaan vain Slave-tietokannoista, pysyvät käytössä olevat Master-tietokannat häiriöttöminä. Tämä minimoi virheiden muodostumisen sekä palvelun häiriintymisen. Käytännössä siis TypingTest-palvelun dumpit tehdään Gigenet 2 -palvelimella ja muut Gigenet 1:llä (kuva 8).

Extranet-palvelimelle tehtiin järjestelmä, joka hakee SaaS-palveluiden varmuuskopiot talteen. Se rakennettiin käyttäen hyväksi Batch-skriptejä, Windows Task Scheduleria sekä WinSCP:tä. Palvelimelle luotiin kaksi erillistä Batch-skriptiä (liite 2), joiden tarkoituksena on ajaa WinSCP-ohjelma, joka hakee varmuuskopiot talteen. Tämä ajastettiin Task Schedulerilla. TypingTest.bat suorittaa WinSCP-ohjelmassa TypingTest.txt:ssä määritellyt komennot. Se ottaa SSH-yhteyden palvelimeen, navigoi varmuuskopiohakemistoon ja synkronoi sen oman varmuuskopiohakemistonsa kanssa (kuva 10). Mahdolliset virheet se kirjoittaa error-lokitiedostoon. Tämän jälkeen se poistaa Extranet-palvelimelta yli 90 päivää vanhat varmuuskopiot. TMOnline.bat toimii täysin samoin. Se käyttää TMOnline.txt -tiedoston komentoja, ja ottaa yhteyden toiselle SaaS-palvelimelle. Sieltä se lataa Extranet-palvelimelle TM Online, TM Online Single, TM LMS sekä AssessTyping -palvelujen tietokantavarmuuskopiot. TM Online -varmuuskopioita pidetään tallessa 90 päivää, ja muita 30. Toinen latausskripti ajetaan kello 9:00 (GMT+2) ja toinen 10:30. Tällöin USA:ssa on jo yö, joten käyttäjiä on vähän, eikä SaaS-palvelimiin kohdistuva ylimääräinen tietoliikenne häiritse palveluja.



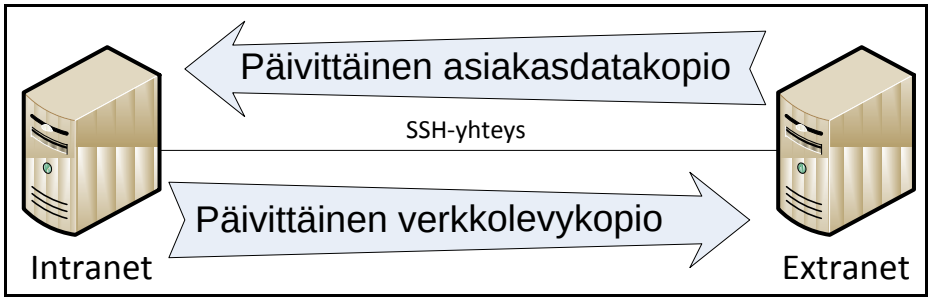
Kuva 10. Varmuuskopioiden synkronointi Batch-skriptillä

Datan arkistointimielessä päätettiin päivittäisistä varmuuskopioista ottaa lisäksi kuukausittaiset kopiot. Tämä toteutettiin Task Schedulerilla Extranet-palvelimella kerran kuukaudessa (joka kuun 30. päivä) ajettavalla Batch-skriptillä (liite 3). Skripti käy jokaisen palvelun varmuuskopiohakemiston erikseen läpi, määrittää uusimman tiedoston ja kopioi sen saman palvelun monthly-hakemistoon. Vanhalle Intranet-palvelimelle taas luotiin skripti, joka WinSCP:tä hyväksi käyttäen lataa nämä kuukausittaiset varmuuskopiot talteen (liite 4). Skripti on rakenteeltaan täysin samanlainen kuin liitteen 2 skriptit. Se ottaa yhteyden joka kuun 30. päivä Intranet-palvelimelta Extranetiin ja synkronoi jokaisen palvelun varmuuskopiohakemiston omien hakemistojensa kanssa. Fyysisesti nämä hakemistot sijaitsevat uudella NAS-laitteella, sillä sen odotettu elinikä on korkeampi kuin Intranet-palvelimen kiintolevyillä.

7.3.2 Tuki-infrastruktuurin varmuuskopiointi

SaaS-palvelujen toiminnan kannalta oleellista on myös palvelujen tuki-infrastruktuurin vikasietoisuus. Vaikka käyttäjädata onkin oleellisin palvelun komponentti, on myös sitä käyttävä ohjelmisto ja konfiguraatiot hyvä varmuuskopioida täydellistä palautumista varten. Näille päätettiin luoda vastaava varmuuskopiointimalli. Lisäksi myös asiakasdata päätettiin liittää samaan malliin, sillä liiketoiminnan jatkuvuuden kannalta myös sen eheys ja toimivuus on oleellista.

Yritykselle luotiin toinen varmuuskopiointijärjestelmä, joka tuli Intranet- ja Extranet-palvelimien välille (kuva 11). Se kopioi Extranet-palvelimella sijaitsevan asiakasdatan eli yrityksen sisäisten palvelujen MySQL-dumpit päivittäin Intranet-palvelimelle. Fyysisesti ne menevät NAS-levylle. Tämän lisäksi Intranet-palvelimelta otetaan päivittäin varmuuskopiot yrityksen verkkolevyistä, joissa sijaitsee liiketoiminnan kannalta kriittistä dataa, sekä ennen kaikkea SaaS-palvelujen konfiguraatiot. Nämä taas siirretään talteen Extranet-palvelimelle.



Kuva 11. Tuki-infrastruktuurin varmuuskopiojärjestelmä

Päivittäiset asiakasdatakopiot luodaan Extranet-palvelimella Batch-skriptillä (liite 5). Se luo MySQL-dumpit kaikista yrityksen sisäisistä palveluista, pakkaa ne 7zip-ohjelmalla ja poistaa yli 14 päivää vanhat varmuuskopiot. Tämän jälkeen ne ladataan Intranet-palvelimelle samalla prosessilla, joka työntää päivittäiset verkkolevykopiot Extranet-palvelimelle (liite 6).

Päivittäiset varmuuskopiot Intranet-palvelimen verkkolevyistä luodaan Batch-skripteillä, jota käyttävät hyväksi Windows Server 2003:n NTBackup-työkalua, 7zip-ohjelmaa sekä WinSCP:tä. Jotta kiintolevy ei täyttyisi heti eikä yrityksen Internet-yhteys ruuhkautuisi, ei täydellistä (full) varmuuskopiota oteta verkkolevyistä joka päivä vaan ainoastaan viikoittain joka lauantai. Differentiaaliset varmuuskopiot taas otetaan joka päivä. Näin voidaan luoda varmuuskopiointiympäristö, josta dataa on helppo palauttaa, mutta se ei vie liikaa levytilaa. Mikäli halutaan palauttaa esimerkiksi keskiviikkona poistetut tiedostot, voidaan tämä tehdä palauttamalla halutut osat edellisen viikonlopun täydestä varmuuskopiosta, jonka päälle voidaan tarpeen vaatiessa palauttaa myös maanantain ja tiistain differentiaaliset varmuuskopiot. Myös täysi palautuminen onnistuu samalla logiikalla. Verkkolevyvarmuuskopioiden lisäksi liitettiin samaan prosessiin myös SVN-varmuuskopio, joka käytännössä sisältää yrityksen tuotteiden uusimmat versiot.

Verkkolevyjen varmuuskopiot jaettiin kahteen eri osaan eli settiin. Toiseen sijoitettiin liiketoiminnan kannalta kriittisimmät hakemistot ja toiseen vähemmän tärkeät tiedostot. Koska Intranet-palvelin on kiinni Internetissä kahden eri yhteyden avulla (Nebula ja Elisa), tällä tavoin saatiin varmuuskopioiden siirtämiseen kuluva aika minimoitua.

Varmuuskopiointi toteutettiin kahdella skriptillä (liite 6). Toinen niistä (backup-tm2.bat) luo varmuuskopiot ja toinen (upload-tm2.bat) siirtää ne Extranet-palvelimelle. Backup-tm2.bat-skripti ajetaan arkipäivisin ”differential”-parametrilla, jolloin se tekee molemmista seteistä differentiaaliset varmuuskopiot ja lauantaisin ”full”-parametrillä, jolloin se taas tekee täydelliset varmuuskopiot. Skripti toimii niin, että ensin se tekee SVN-varmuuskopion, minkä jälkeen se suorittaa NTBackup-ohjelman ensimmäiselle setille, joka luo bkf-muotoiset varmuuskopiot määritellyistä levyistä ja hakemistoista. Tämän jälkeen bkf-tiedosto pakataan ja kryptataan 7zip-ohjelmalla. Kun tämä on valmis, kutsutaan upload-tm2.bat-skriptiä ensimmäisen setin parametreille ja siirrytään suorittamaan vastaavaa NTBackup & 7zip -prosessia toiselle setille. Upload-tm2.bat käynnistyy taustalle, ottaa WinSCP:n avulla yhteyden Extranet-palvelimelle ja siirtää luodun bkf.7z -tiedoston talteen. Se poistaa myös yli 7 päivää vanhat varmuuskopiot Intranet-palvelimelta. Kun toisen setin luomisprosessi on valmis, kutsutaan upload-tm2.bat -skriptiä uudelleen toisen setin parametreilla, minkä jälkeen myös se siirretään Extranet-palvelimelle. Aivan lopuksi ladataan vielä Extranet-palvelimen MySQL-dumpit talteen Intranet-palvelimelle.

Koska Extranet-palvelimella on kaksi eri IP-osoitetta, siirretään kukin varmuuskopiosetti eri osoitteeseen. Intranet-palvelin taas on kiinni Load Balancer -laitteessa, johon on määritelty, että toinen Extranet-palvelimen IP:istä putkitetaan Elisan Internet-yhteyden läpi ja toinen Nebulan. Tällä tavoin molemmat setit saadaan siirrettyä samanaikaisesti eri yhteyksiä hyväksi käyttäen.

Verkkolevyistä otetut täydelliset varmuuskopiot siirretään arkistointimielessä kuukausittain erilliseen hakemistoon (liite 3). Tavallisesti täydellisiä varmuuskopioita pidetään tallessa 60 päivää, ja differentiaalisia taas 7 päivää. Tämä siivous on toteutettu erillisellä päivittäin ajettavalla skriptillä (liite 7), joka poistaa liian vanhat varmuuskopiot molemmista seteistä. Arkistointi tehdään siksi, että on mahdollista, että jokin tärkeä tiedosto täytyy palauttaa jopa kuukausien päästä.

7.3.3 Virheentunnistus ja ilmoitukset

Uuden varmuuskopiointimallin myötä täytyi myös virheentunnistusjärjestelmää päivittää. Periaatteena oli, että virheentunnistuksen ja ilmoituksien lähettämisen täytyi olla ennen kaikkea luotettavaa. Tämän takia ne toteutettiin lähinnä itse tehdyillä skripteillä, jotta tiedetään tarkalleen, mitä mikäkin tekee, sekä jo hyväksi havaitulla kolmannen osapuolen valvontapalvelulla. Virheet jaettiin kahteen eri osa-alueeseen: varmuuskopiointivirheisiin ja palveluvirheisiin. Jälkimmäinen tarkoittaa käytännössä käyttökatkosta tai vakavaa uhkaa palvelun infrastruktuurille. Vakavia virheitä varten luotiin uusi sähköpostitili, johon ilmoitukset tulisivat. Täten palvelun kaatumisesta tiedettäisiin heti.

Palveluvirheitä monitoroi edelleen SiteUptime-palvelu (kuva 12). Se asetettiin tarkistamaan palveluiden saatavuus viiden minuutin välein. Tähän kuuluivat lisäksi myös yrityksen sisäisessä käytössä olevat palvelut, mukaan lukien uusi Extranet-palvelin. Extranet-palvelimen palveluntarjoajan hallintapaneelista asetettiin lisäksi vielä erillinen tarkistin päälle, joka lähettää sähköposti-ilmoituksen, mikäli palvelin ei vastaa. SiteUptime on myös kätevä työkalu uptime-statistiikan valvontaa varten.

Monitor Status				
Monitor Name	**Service**	**Last check**	**Uptime**	**Current Status**
www.TypingMaster.com	http	March 15, 2012 at 12:14	99.879%	Ok
www.typingtest.com	http	March 15, 2012 at 12:14	99.701%	Ok
	ping	March 15, 2012 at 12:19	99.603%	Ok
online.typingmaster.com	http	March 15, 2012 at 12:19	99.982%	Ok
online3.typingmaster.com	http	March 15, 2012 at 12:19	99.964%	Ok
	http	March 15, 2012 at 12:14	99.624%	Ok
	https	March 15, 2012 at 12:18	99.988%	Ok
www.assesstyping.com	http	March 15, 2012 at 12:18	99.841%	Ok

Kuva 12. SiteUpTime-monitorointipalvelu

SiteUptime varoittaa kyllä nopeasti virheistä, mutta sen piiriin sisältyvät vain katastrofaaliset palvelukatkokset. Jos mahdollista, halutaan katkoksesta tietää jo ennen sen syntymistä. Tämän takia molemmille uusille SaaS-palvelimille luotiin järjestelmä valvomaan palvelimien RAID-pakan toimintaa sekä palvelimen lämpötiloja. Nämä toteutettiin

Cronissa ajettavilla skripteillä, joiden pohja haettiin Internetistä [26]. Ensimmäinen skripti (liite 8) ajetaan viiden minuutin välein, ja se tarkistaa RAID Controllerin omaa työkalua hyväksi käyttäen RAID-pakan tilan. Mikäli virheitä ilmenee, lähettää se varoituksen uuteen sähköpostiosoitteeseen. Toinen skripti (liite 9) taas ajetaan minuutin välein ja sen tehtävänä on valvoa palvelinlaitteiston omien sensoreiden avulla palvelimen lämpötiloja. Se tarkistaa neljän eri CPU-sensorin lämpötilat sekä kahden tuulettimen toiminnan. Jos lämpötila ylittää arvon 80 °C tai jos tuuletin pysähtyy, niin sähköposti-ilmoitus lähetetään luotuun osoitteeseen. Tällä tavoin mikäli SaaS-palvelinta kiintolevyissä tai tuulettimissa ilmenee häiriöitä, tai jos lämmöt nousevat liian korkeiksi, saadaan siitä tieto heti.

Varmuuskopiointivirheiden tunnistus integroitiin hyvin pitkälti uusin skripteihin, jotta järjestelmä pysyisi yksinkertaisena. Niiden vakavuus ei ole samaa luokkaa kuin palveluvirheiden, joten ne eivät vaadi välittömiä toimenpiteitä. Tästä huolimatta myös varmuuskopioinnin yhteydessä ilmenevät virheet täytyy tietysti korjata, jotta varmuuskopioitu data on ehjää ja yhtenäistä. Virheentunnistus luotiin jokaiseen kriittiseen varmuuskopiointiskriptiin.

Skripti (liite 1), joka luo SaaS-palvelujen MySQL-dumpit, on kaikkein kriittisin, sillä käyttäjädatan täytyy olla kunnossa. Tämän takia siihen tehtiin järjestelmä, joka vertaa edellisen päivän dumpin kokoa juuri tehtyyn dumppiin. Mikäli uusi dumppi on samankokoinen tai pienempi, lähetetään siitä sähköposti-ilmoitus. Ilmoituksen lähettäminen tehdään yksinkertaisella Sendmail-ohjelmalla. Tämä tehtiin siksi, että yrityksen SaaS-palvelut on rakennettu niin, että käyttäjädataa kasvaa joka päivä. Mikäli näin ei käy, on jotain pielessä.

Kaikissa WinSCP:tä käyttävissä skripteissä (liitteet 2, 4, 6) on parametri, joka kirjoittaa mahdolliset virheet lokitiedostoon, mikäli ohjelma palauttaa virhekoodin 2 eli sen ajo epäonnistuu. Tästä ei tule ilmoitusta, mutta virheen sattuessa tätä kautta saadaan helposti selville, mitä on tapahtunut, mikäli lataus- tai siirtoprosessi epäonnistuu. Erityisen tehokas virheentunnistusjärjestelmä tehtiin skriptiin (liite 6), jonka vastuulla on verkkolevyvarmuuskopioiden luominen ja siirto. Vaikka tämä ei olekaan SaaS-palvelujen vikasietoisuuden kannalta kriittisin prosessi, on se sitä yrityksen muun liiketoiminnan kannalta.

Backup-tm2.bat-skriptiin tehtiin järjestelmä, joka mahdollisen virheen sattuessa lähettää virheilmoituksen Sendmail-ohjelman avulla. Se voi tapahtua missä vain kohti varmuuskopion luomisprosessia (SVN, NTBackup, 7zip). Virheviestissä kerrotaan, että nimenomaan luomisprosessissa on tapahtunut virhe. Vastaava järjestelmä tehtiin myös upload-tm2.bat-skriptiin, mutta se asetettiin odottamaan 10 minuuttia mahdolliseen virheen sattuessa ja sitten yrittämään uudestaan. Mikäli siirto ei vieläkään onnistu, yrittää skripti sitä neljän tunnin ajan kymmenen minuutin välein. Tämän jälkeen se lähettää sähköposti-ilmoituksen ja sulkee itsensä. Tämä tehtiin siksi, että verkkolevyvarmuuskopiot ovat kohtalaisen suuria tiedostoja (parhaillaan kaksi gigatavua), joten niiden siirto on erityisen herkkä verkkovirheille. WinSCP osaa automaattisesti jatkaa siirtoprosessia siitä mihin se jäi, joten jos verkkoyhteys katkeaa vaikka puoleksi tunniksi, saadaan varmuuskopio silti siirrettyä ilman, että siitä tulee virheilmoitus. Mikäli neljän tunnin jälkeenkään siirto ei ole vielä onnistunut, lähetetään ilmoitus, jolloin se vaatii todennäköisesti toimenpiteitä.

Mahdollinen palveluvirhe vaatii pääsääntöisesti aina välitöntä toimintaa. Varmuuskopiointivirhe taas ei ole niin kriittinen, vaikka sekin täytyy tietysti saada korjattua.

7.3.4 Muut tekijät

Kuten on käynyt ilmi, ei yrityksen SaaS-palveluiden toimintavarmuus ole kovinkaan yksipuolinen asia. Se on useisiin eri komponentteihin perustuva kokonaisuus, joista osa on kriittisiä ja osa taas lähinnä tärkeitä. Varsinaisen palvelu- ja verkkoinfrastruktuurin sekä varmuuskopioinnin lisäksi toimintavarmuuteen vaikuttavat vielä eräät muutkin seikat. Tärkeimmät näistä ovat ulkoistetut DNS- ja sähköpostipalvelut. Näistä toinen liittyy suoraan palveluiden saatavuuteen ja toinen taas lähinnä asiakashallintaan. Molempien varsinainen toimivuus on täysin palveluntarjoajien vastuulla.

Vaikka yritys onkin ostanut Domain-nimiään useilta eri palveluntarjoajilta, on niiden hallinta keskitetty samalle DNS-palveluntarjoajalle nimeltä DNSMadeEasy. Tänne on määritelty kaikkien tässä työssä mainittujen palveluiden Domain-tietueet. Jotta oikea palvelin löytyy oikeasta paikasta, on oleellista, että DNS toimii. Tarpeen tullen saman palvelun kautta voidaan määrittää myös toissijaisia IP-osoitteita samalle DNS-nimelle,

mutta yrityksen palveluiden vikasietoisuus ei ole vielä niin automatisoitu, että tälle olisi tarvetta.

Liiketoiminnan jatkuvuuden kannalta erityisen tärkeä on yrityksen sähköpostipalvelu, joka on ostettu palveluntarjoajalta nimeltä Intermedia. Kaikki yrityksen sähköpostiosoitteet on luotu täällä, mukaan lukien SaaS-palveluiden ja sisäisten palveluiden automaattiset mailerit. Kaikki yrityksen palvelut ovat luonteeltaan sellaisia, että kommunikaatio asiakkaiden kanssa tapahtuu sähköpostitse, suurin osa automaattisesti. Tämän takia on tärkeää, että sähköpostipalvelut toimivat.

Toimintavarmuus ei kuitenkaan rajoitu vain näihin tekijöihin, vaan siihen liittyy lisäksi lukemattomia pienempiä asioita. Esimerkiksi palveluiden ja teknisen infrastruktuurin dokumentointi on tärkeää. Tämä on yrityksessä toteutettu sisäisellä Wiki-sivulla. Lisäksi sisäisten palvelujen käyttäjätunnukset ja salasanat on hyvä olla tallessa sekä turvassa. Se, missä toimintavarmuuden ja palvelun pyörittämisen raja menee, on todella häilyvä. Voidaankin sanoa, että kaikki palvelun toiminnalliset ja hallinnolliset komponentit liittyvät jossain määrin sen toimintavarmuuteen. Tämän takia tässä työssä keskitytään vain olennaisimpiin.

8 Riskit ja uhat

Vaikka SaaS-järjestelmän toimintavarmuus ja vikasietoisuus saatiinkin uuden verkkoinfrastruktuurin ja varmuuskopiointijärjestelmän avulla aivan uudelle tasolle, kohdistuu palveluihin edelleen merkittäviä uhkia, jotka olisi hyvä tiedostaa. Luvussa 5 käsiteltiin erilaisia SaaS-palvelujen ongelmia ja syitä palvelukatkoksiin. Osa näistä pätee myös kyseessä oleviin yrityksen palveluihin. Vaikka toimintavarmuutta saataisiinkin parannettua, on useimpia vakavia uhkia lähes mahdotonta eliminoida. Niihin voidaan vain varautua. Ei siis ole sattumaa, miksi tässä työssä keskitytään suurissa määrin varmuuskopiointiin ja palautumiseen. Niiden avulla saadaan luotoa vikasietoinen järjestelmä, joka on tehokkain ase SaaS-palveluihin kohdistuvia uhkia vastaan.

Kuten kappaleessa 9.3.3 todettiin, voidaan mahdolliset virheet jakaa itse palveluun kohdistuviin virheisiin sekä varmuuskopiointiin kohdistuviin virheisiin. Samoin on riskien

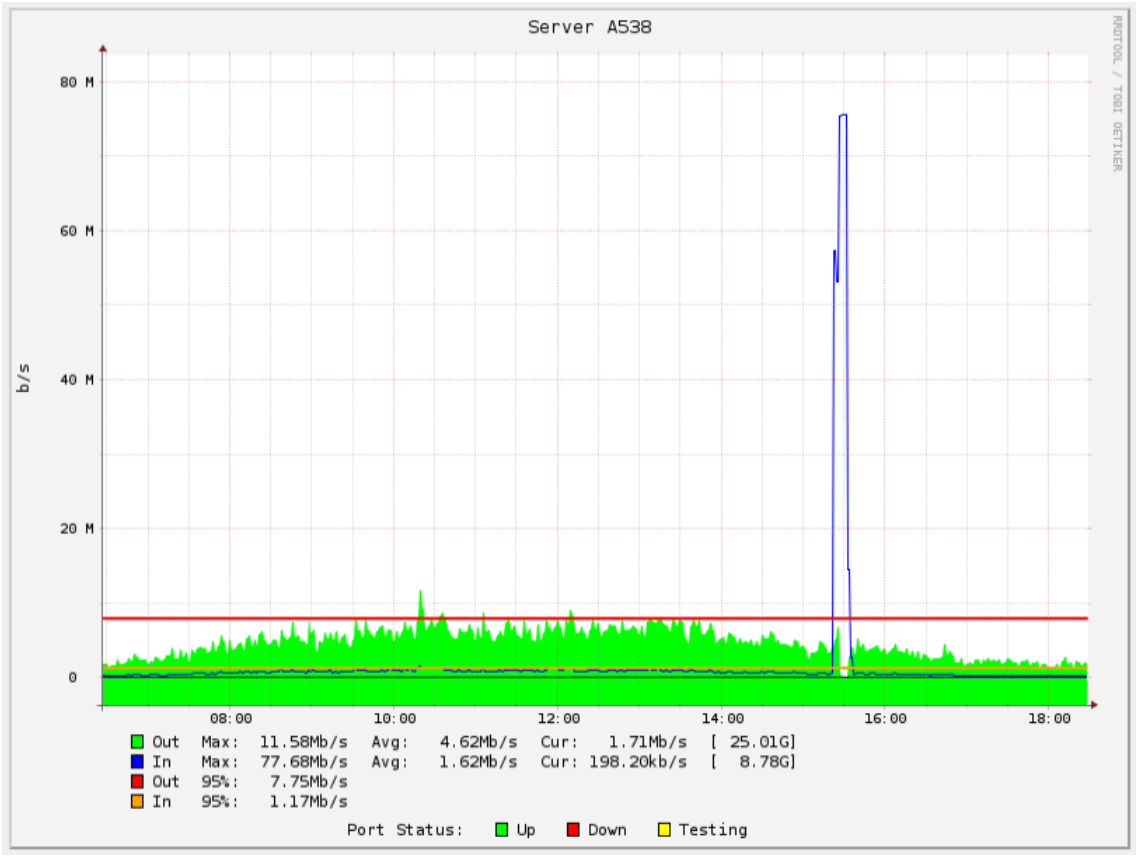
kanssa. Vaikka useimmat suuret virheet tapahtuessaan aiheuttavatkin sekä palveluiden sekä varmuuskopioiden katkoksen, ei aina näin ole. Esimerkiksi jonkin tukiverkon komponentin kaatuminen on uhka ainoastaan varmuuskopioinnin toimivuudelle. Toisaalta taas ohjelmistovirhe vaikuttaa tapahtuessaan hyvin todennäköisesti vain palvelun toimintaan.

Koska yrityksen SaaS-palvelut ovat olleet toiminnassa jo useita vuosia, ei ohjelmistouhkia pitäisi juurikaan olla. Pieniä päivityksiä tehdään aina välillä, joiden myötä pieniä bugeja saattaa ilmetä, mutta ne korjataan usein hyvin pikaisesti.

Palautumisprosessiin saatetaan ryhtyä useista eri syistä. Näitä kuitenkin kaikkia yhdistää se piirre, ettei alkuperäistä palvelinta saada järkevällä aikavälillä enää toimintakuntoon. Tällaisia uhkia ovat muun muassa kiintolevyn hajoaminen, järjestelmälaajuinen virhe ja erilaiset laajamittaiset katastrofit. Käytännössä siis useimmat vakavat fyysiset virheet aiheuttavat palautumiseen ryhtymisen.

Luvaton murtautuminen on yrityksen SaaS-palveluille uhka siinä missä muillekin palveluntarjoajille. Murtautumista ehkäistään tehokkaasti käyttämällä SaaS-palvelimissa Linux-ohjelmistoja, joiden tietoturva on todettu erinomaiseksi. Tarkoin rajatut iptables-säännöt estävät luvattomat yhteydet, eikä ohjelmistoissa käytetä oletusarvoisia portteja. Poikkeuksena on Tomcat-webbisovellus, sillä se on palvelimen pääasiallinen palveluohjelmisto.

Palvelunestohyökkäyksiä ei valitettavasti pystytä täysin ehkäisemään mitenkään. Menneisyydessä niistä ei ole juurikaan ollut riesaa, joten niiltä suojautumiseen ei ole panostettu. Tätä työtä tehdessä kävikin niin, että eräänä perjantaina toinen SaaS-palvelimista joutui palvelunestohyökkäyksen kohteeksi (kuva 13). Kuten kuvasta näkyy, kohdistui palvelimeen 10 minuutin aikavälillä lähes 80 Mb/s edestä sisääntulevaa liikennettä. Varmuutta tämän lähteestä ei ole, mutta se todennäköisesti tuli useista eri paikoista (DDoS). Muutaman minuutin sisällä palvelin meni täysin jumiin, eikä se vastannut enää mihinkään. Ainut tehtävissä oleva asia oli palvelimen täydellinen uudelleenkäynnistys. Tämän jälkeen huomattiin, että myös replikaatio oli mennyt solmuun, ja se jouduttiin käynnistämään uudestaan.



Kuva 13. DDoS-hyökkäys palvelimelle

Hyökkäykseltä olisi voitu suojautua palveluntarjoajan DDoS Protection -palvelun avulla, mutta se on yrityksen tarpeisiin liian kallis ja järeä. Vastaavaa suojausta voidaan toteuttaa myös ohjelmistotasolla esimerkiksi psad [27] ja/tai fwsnort [28] -työkalujen avulla. Näiden implementointi SaaS-palvelimiin on hyökkäyksen jälkeen suunnitteilla.

9 Palautuminen

Täydellistä toimintavarmuutta ei voida koskaan taata. Tämän takia on oleellista suunnitella myös palvelun palautuminen kunnolla. Tällöin ajattelussa siirrytään toimintavarmuudesta vikasietoisuuteen ja redundanssiin. Käytännössä tämä tarkoittaa sitä, miten nopeasti ja tehokkaasti yrityksen palvelut saadaan katkon jälkeen taas toimintakuntoon. Varmuuskopioista ei ole mitään hyötyä, jos niitä ei voida palauttaa.

Kuten kappaleessa 9 todettiin, kohdistuu yrityksen palveluihin monenlaisia ja monitasoisia eri uhkia. Iso osa näistä ei vaadi yrityksen henkilöstöltä minkäänlaisia toimen-

piteitä, mutta kaikkein vakavimmat vaativat välitöntä toimintaa. Yrityksen palveluiden tapauksessa palautumisen piiriin kuuluvat juuri nämä toimenpiteitä vaativat uhat. Automaattista tai palveluntarjoajan vastuulle kuuluvaa palautumista ei tarvitse tai kannata suunnitella, sillä se olisi resurssien haaskausta. Automaattiseksi palautumiseksi voidaan laskea useimmat pienemmät riskit, kuten esimerkiksi hetkellinen verkkoyhteyden katkeaminen. Vaikka ilmoitus asiasta tulee, ei se vaadi toimenpiteitä.

Koska yrityksen palveluiden failover jätettiin jossain määrin manuaaliseksi prosessiksi, ei täydellinen palautuminen toimi automaattisesti. Tämä tehtiin siksi, että päätös palautusprosessin suorittamisesta tulee olla yrityksen teknisillä asiantuntijoilla. Jos tulee esimerkiksi lyhyt palvelukatkos, ei se vielä vaadi palautusta. Tällä tavoin vastuun ollessa henkilöstöllä, tiedetään tarkalleen, mitä tapahtui ja miten siihen tulee reagoida. Lisäksi täysin automaattisen failover-järjestelmän luominen nykyisillä resursseilla olisi todella haastavaa, ellei jopa lähes mahdotonta. Manuaalisuus ei kuitenkaan tarkoita sitä, että palautuminen tai korjaus pitäisi tehdä tilannekohtaisesti improvisoiden. Päinvastoin siihen tulee varautua mahdollisimman hyvin niin, että suuret palautusoperaatiot vaativat vain muutaman toimenpiteen. Pienempiä virheitä tietysti korjataan tilanteen mukaan, sillä kaikkeen ei ole mahdollista varautua.

9.1 Varautuminen

Vaikka pyrkimyksenä on, ettei palautusta tarvitsisi koskaan tehdä, tulee se silti vääjäämättä joskus vastaan. Tämän takia siihen tulee varautua hyvin. On erityisen tärkeää että palveluiden palautus toimii saumattomasti ja ilman ongelmia. Hyvään toimintamalliin kuuluukin, että mahdollinen palauttaminen suunnitellaan, dokumentoidaan ja testataan tarkoin.

Kaikki palautusprosessit eivät ole massiivisia toimenpiteitä. Halutessa voidaan palauttaa esimerkiksi vain edellisen päivän tietokanta, mikä ei vaadi minkäänlaista suunnitelmaa. Tällöin haluttu dumppi haetaan palvelimelle, Tomcat pysäytetään hetkeksi, dumppi ajetaan sisään tietokantaan ja Tomcat käynnistetään jälleen.

Jotta täysin palautuminen olisi mahdollisimman vaivatonta, tehtiin molempiin SaaS-palvelimiin erinäisiä toimenpiteitä. Molemmilla palvelimilla on identtinen verkkoinfra-

struktuuri. Lisäksi kaikki palvelukomponentit pidettiin molemmilla palvelimilla samoina (kuva 3). Molemmissa on siis täysin sama ohjelmisto sekä käyttäjädata ja lähes samat palvelinkonfiguraatio. Merkittävin ero on, että toisella palvelimella tietokannat ovat Slave-tilassa, joten niihin ei kirjoiteta mitään. Lisäksi Tomcatin Virtual Hostit ovat molemmissa palvelimissa identtiset, mutta toisessa on osa hosteista kommentoitu pois. Tämä tarkoittaa sitä, että Tomcat-konfiguraatiot ovat valmiina mutta eivät käytössä. Kaiken lisäksi kaikki palvelinkonfiguraatiot löytyvät myös yrityksen verkkolevyltä.

Luodun järjestelmän tapauksessa ei voida puhua Cold Standby eikä Hot Standby -tiloista, vaan jostakin siltä väliltä. Kaikki palvelut löytyvät molemmilta palvelimilta, mutta palautuksen yhteydessä ne pitää aktivoida manuaalisesti. Tämä on kuitenkin yksinkertainen toimenpide.

Kaikki varmuuskopiot pidetään tallessa niin kauan, kuin niistä voi olla palautuksen kannalta hyötyä. Tämän jälkeen niistä otetaan kuukausittainen arkistointikopio. Varmuuskopioiden säilytysajat näkyvät taulukossa 3.

Taulukko 3. Varmuuskopioiden säilytysajat ja -paikat

	Intranet		**Extranet**	
	Päivittäiset	**Kuukausittaiset**	**Päivittäiset**	**Kuukausittaiset**
Online-palvelut	—	loputtomasti	30 päivää	loputtomasti
TypingTest	—	loputtomasti	90 päivää	loputtomasti
Verkkolevyt	7 päivää	loputtomasti	Diff. 7 päivää Full 60 päivää	loputtomasti
Sisäiset palvelut	loputtomasti	—	14 päivää	—

9.2 Palautusprosessi

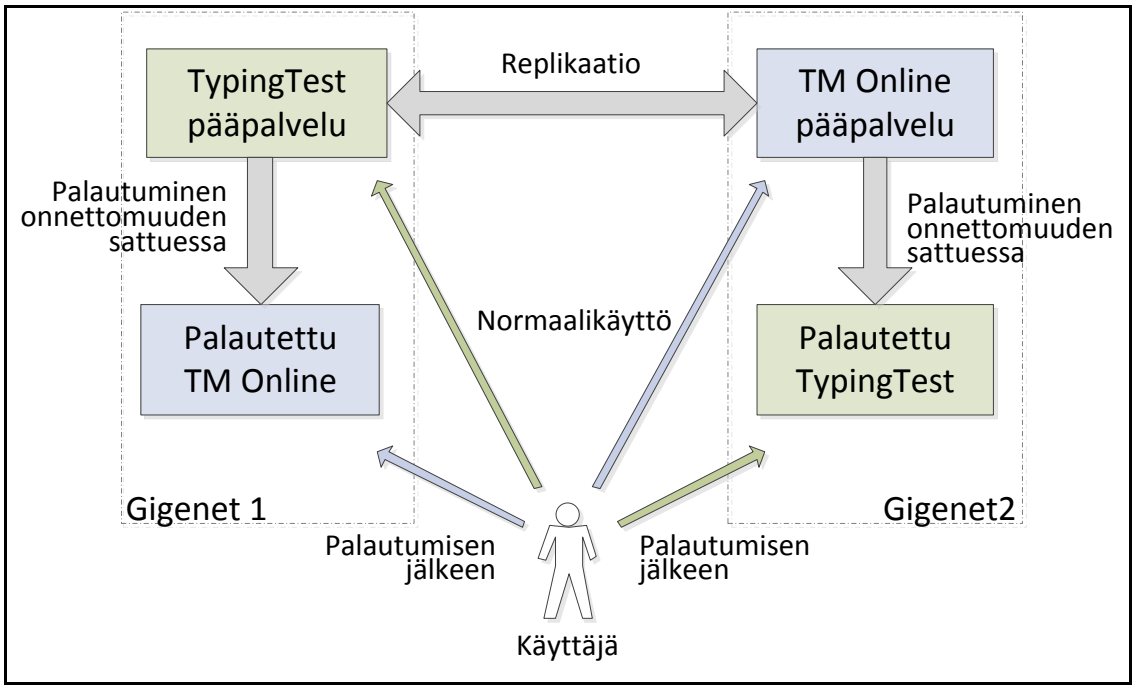
Mikäli tarve palautukselle tulee, täytyy suoritettava prosessi olla valmiiksi suunniteltu ja testattu. Näin tehtiinkin heti, kun varmuuskopiointijärjestelmä saatiin toimimaan. Palautusprosessi suunniteltiin sekä palveluille, että tuki-infrastruktuurille. Palveluiden palautuksella tarkoitetaan käytännössä SaaS-palveluja, ja tuki-infrastruktuurilla taas verkkolevyvarmuuskopioita ja yrityksen sisäisiä palveluja.

9.2.1 Palvelujen palautus

SaaS-palveluiden palautus voi tarkoittaa joko koko palveluinfrastruktuurin palautusta eli palvelimen uudelleenasennusta tai sitten pelkästään käyttäjädatan palautusta varmuuskopioista. Palvelujen palauttaminen voidaan jakaa myös eri tasoihin riippuen virheen tai palvelukatkoksen vakavuudesta. Esimerkiksi ohjelmiston kaatuminen ei sinänsä vaadi suunniteltua palautumisprosessia, vaikka se toimenpiteitä yleensä vaatiikin. Monet pienemmät ongelmat ja ohjeet niistä selviämiseen on dokumentoitu yrityksen Wiki-sivulle.

Vakavin skenaario joka johtaa palautusprosessiin, on että koko palvelin menee toimintakyvyttömäksi. Tällöin kyseisellä palvelimella pyörineet SaaS-palvelut täytyy saada mahdollisimman nopeasti jälleen käyttöön (kuva 14). Tätä varten asennettiin MySQL-replikaatio, jotta palvelun palauttaminen toiselle palvelimelle olisi nopeaa ja sujuvaa. Mikäli toiselle palvelimelle sattuu jotain, täytyy ensin määrittää, voiko palvelun saada toimimaan vielä siinä samassa palvelimessa kohtuullisen lyhyellä aikavälillä. Mikäli näin ei ole, siirrytään palauttamaan kyseisellä palvelimella pyörineitä palveluita toiselle palvelimelle. Tällöin suoritetaan seuraavat toimenpiteet (dokumentoitu tarkemmin yrityksen Wiki-sivulle):

1. Varmistetaan, ettei kaatunut palvelin ole käytössä tai häiritse palvelujen siirtoa.
2. Kaatuneiden palvelujen tietokannat otetaan pois Slave-tilasta.
3. Siirrettävien palveluiden Virtual Hostit aktivoidaan Tomcatissä.
4. Tomcat käynnistetään uudestaan.
5. Palveluiden DNS muutetaan osoittamaan uudelle palvelimelle.
6. Palvelun toimivuus testataan läpikotaisin.



Kuva 14. SaaS-palvelujen palautussuunnitelma

Kun kaikki palvelut ovat jälleen toimintakunnossa, siirrytään korjaamaan rikki mennyttä palvelinta. Useimmiten tämä jää tosin palveluntarjoajan vastuulle. Tämä prosessi ei ole enää kovinkaan kiireinen, vaikka muutaman päivän sisään sekin pitäisi tietysti saada hoidettua, jotta palautuminen takaisin lähtötilanteeseen saadaan suoritettua täydellisesti. Kun palvelin on taas toimintakunnossa, asennetaan palveluinfrastruktuuri siihen uudestaan. Tarkat ohjeet tätä varten on dokumentoitu yrityksen Wiki-sivulle. Kun kaikki on asennettu, siirretään sopivan hetken tullen palvelimen omat palvelut takaisin siihen. Käytännössä tämä tarkoittaa vain sitä, että palvelut pysäytetään toisella palvelimella, niistä otetaan MySQL-dumpit, dumpit importoidaan uudelle palvelimelle, MySQL ja Tomcat käynnistetään ja DNS asetetaan ohjaamaan jälleen oikealla palvelimelle.

Mikäli käyttäjädataa täytyy palauttaa esimerkiksi korruptoitumisen takia, voidaan data hakea joko toiselta SaaS-palvelimelta tai Extranet-palvelimelta. Se kumpaa käytetään, on tapauskohtaista. Mikäli palvelu menee rikki ja asiakasdatan palautukseen joudutaan, olisi ensisijaisesti optimaalista käyttää toisella SaaS-palvelimella olevaa replikoitua Slave MySQL -tietokantaa, sillä se sisältää kaikkein uusimman datan. Tällöin mitään dataa ei pääse häviämään.

Jos on tilanne, ettei replikoitua tietokantaa voida käyttää, täytyy uusin varmuuskopio hakea Extranet-palvelimelta. Jos esimerkiksi koko tietokanta on korruptoitunut, ei sen replikaatiota tietenkään voida käyttää. Päivittäistä varmuuskopiota käyttämällä menetetään jonkin verran dataa, mutta se on väistämätöntä.

Käyttäjädatan palautus tapahtuu yksinkertaisesti joko hakemalla varmuuskopio Extranet-palvelimelta tai ottamalla dumppi Slave-tietokannasta (riippuen siitä, kumpaa käytetään) siirtämällä se palautettavalle palvelimelle ja importoimalla se.

9.2.2 Tuki-infrastruktuurin palautus

Tuki-infrastruktuurin palautus voi tarkoittaa joko verkkolevyjen (tai tiedostojen) palautusta tai sitten yrityksen sisäisten palvelujen asiakasdatan palautusta. Verkkolevyjen palautusta varten luotiin yksityiskohtaiset ohjeet (liite 10), joiden avulla voidaan palauttaa joko koko levy tai yksittäisiä tiedostoja. Käytännössä halutut varmuuskopiot haetaan Extranet-palvelimelta Intranet-palvelimelle, ne puretaan ja palautetaan NTBackup-työkalulla. Jopa yksittäisten tiedostojen palautus voidaan suorittaa todella tarkasti, sillä täysiä verkkolevyvarmuuskopioita säilytetään todella kauan. Tätä ei kuitenkaan aina tarvitse edes tehdä, sillä palvelimen käyttöjärjestelmä tukee toiminnassa olevan Shadow Copy Servicen ansiosta Previous Version -ominaisuutta [29]. Tämän ansiosta poistettuja tai muutettuja tiedostoja ja hakemistoja voidaan palauttaa helposti lähimenneisyydestä.

On todella epätodennäköistä, että asiakasdataa täytyy palauttaa. Tästä huolimatta siihenkin on varauduttu. Se onnistuu yksinkertaisuudessaan niin, että pakatut MySQL-dumpit haetaan Intranet-palvelimelta Extranet-palvelimelle, jonka jälkeen ne importoidaan haluttuihin tietokantoihin. Jos itse Extranet-palvelimelle tapahtuu jotain, voidaan tarvittaessa kaikki yrityksen sisäiset palvelut saada pyörimään myös Intranet-palvelimella.

9.3 Testaus

Kuten hyvien tapojen mukaista on, täytyy palautuminen myös testata. Mikäli mahdollista, täytyy se myös tehdä usein ja eri tavoin. Monet surullisenkuuluisat tapaukset muis-

tuttavat siitä, että varmuuskopioista ei ole mitään hyötyä, mikäli ne eivät toimikaan [17]. Palautumisen testaus suoritettiin jokaiselle tehdylle varmuuskopiointijärjestelmälle. Käytössä olevien palvelujen varmuuskopioiden toimivuus on jossain määrin haastavaa todentaa, sillä ne vaativat toimiakseen täyden palveluinfrastruktuurin. Nämä kuitenkin testattiin eräässä virtuaalikoneessa. Palveluista valittiin sattumanvaraisia varmuuskopioita, ne ladattiin virtuaalipalvelimelle ja importoitiin tietokantoihin. Tämän jälkeen todettiin, että palvelut toimivat virtuaalikoneella kuten pitääkin, ja data oli sellaista kuin kuuluu.

Myös verkkolevytiedostojen palautus testattiin. Tämä tapahtui liitteessä 10 määritellyillä ohjeilla. Kaikki toimi kuten kuuluukin.

Vaikka luotu järjestelmä onkin verrattain yksinkertainen ja vikasietoinen kaikkine ilmoituksineen, on aina mahdollista että ohjelmistossa menee jotakin pieleen ja varmuuskopiot korruptoituvat. Tämän takia on suositeltavaa, että varmuuskopioiden toimivuus todennetaan vähintään muutaman kuukauden välein.

10 Arvio

Asetettuihin tavoitteisiin yrityksen SaaS-palvelujen toimintavarmuuden parantamiseksi päästiin. Työssä toteutettujen uudistusten jälkeen se on kiitettävässä kunnossa. Työn aikana kävi ilmi, miten huonosti palvelujen dataa alun perin hoidettiin. Vaikka se ennen tiedostettiinkin puutteeksi, ei sen vakavuus ollut selvillä. Tästä kertoo muun muassa luvussa 6.1 läpi käyty onnettomuus, joka suositulle SaaS-palvelulle olisi nykypäivänä anteeksiantamaton. Tämä ymmärrettiin jo hyvissä ajoin ja virheistä opittiin, minkä takia palvelujen toimintavarmuutta päädyttiin parantamaan.

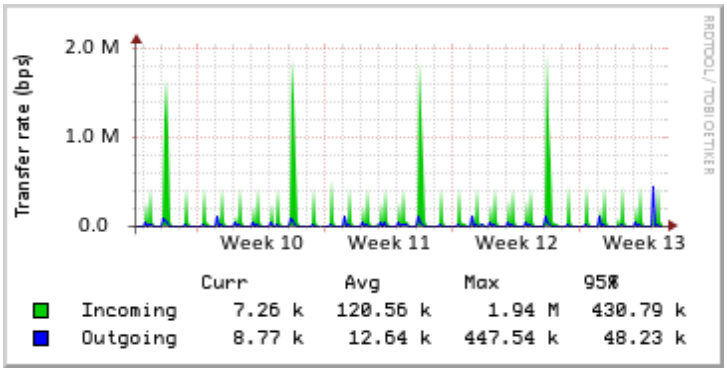
Yrityksen resurssien rajallisuudesta huolimatta saatiin uudesta varmuuskopiointijärjestelmästä tehtyä todella hyvä. Se on kokonaisuutena yksinkertainen, mutta myös vikasietoinen. Luotu varmuuskopiointijärjestelmä on myös helposti hallittava, sillä se on kokonaan itse rakennettu. Toisaalta koska se tehtiin lähinnä avoimen lähdekoodin pikkuohjelmia hyväksi käyttäen, vaaditaan järjestelmän hallintaa varten jonkin verran asi-

antuntemusta. Periaatteena kuitenkin on, ettei järjestelmää juurikaan tarvitsisi hallita, vaan se toimisi omillaan.

Vaikka palvelujen vikasietoisuus ei olekaan täysin automaattinen prosessi, saatiin sitä parannettua merkittävästi. Niin kauan kuin yrityksellä on tietoteknistä työvoimaa, pikkuvioista selvitään helposti ja palveluja valvotaan jatkuvasti. Tähän nähden luotu järjestelmä on erittäin hyvä apuväline, jolla varaudutaan mahdollisiin virheisiin. Palautuminen hälytyksen jälkeen on nopeaa, sillä koko palautusprosessi on suunniteltu valmiiksi.

Varmuuskopiointijärjestelmä on nykyisessä muodossaan toiminut tätä kirjoitettaessa noin kaksi kuukautta, eikä sinä aikana ole ilmennyt ylitsepääsemättömän vakavia puutteita järjestelmässä tai yrityksen palveluissa. Joitakin pikkuvikoja on ilmennyt, mutta ne on korjattu heti. MySQL-replikaatio on erityisen herkkä vioille, mutta siihen kohdistuvat viat on usein helppo korjata, eikä se aiheuta minkäänlaista datan häviämistä.

Sekä palvelujen että varmuuskopiointijärjestelmän toimintaa valvotaan työssä esiteltyjen periaatteiden mukaisesti. Kuvassa 15 esitetään Extranet-palvelimen verkkoliikennestatistiikkaa, jossa ajastetut varmuuskopiot näkyvät selvästi. Jokaisena viikonloppuna näkyy selkeästi verkkolevyjen täydellisten varmuuskopioiden siirto, ja pienempi päivittäinen sisääntuleva liikenne tarkoittaa lähinnä SaaS-palveluiden varmuuskopioita.



Kuva 15. Extranet-palvelimen verkkoliikennestatistiikkaa

Erityistä huomiota työn teon aikana kohdistui SaaS-palvelujen tietoturvaan. Se on pyritty pitämään yksinkertaisena, mutta ehkä jopa liian yksinkertaisena. Pelkkä iptables-palomuuriohjelma ja palvelimilla käytetyt hyvien tapojen mukaiset tietoturvaperiaatteet

eivät välttämättä estä nykyaikaista tunkeutujaa. Täten esimerkiksi hyökkäyksiä pitäisi pyrkiä estämään aktiivisesti eikä vain reaktiivisesti. Kuten luvussa 8 todettiin, on tämä jo suunnitteilla.

Yrityksen SaaS-palvelujen data saa nyt ansaitsemaansa huomiota. Se säilötään hyvin, ja myös palautuminen onnistuu tarpeen vaatiessa vaivattomasti. Seuraavaksi pitäisikin pyrkiä siihen, ettei palautukseen jouduta. Miten tämä käytännössä tehdään, selviää myöhemmin.

Lähteet

1. Software as a service. 2012. Wikipedia tietosanakirja. <http://en.wikipedia.org/wiki/Software_as_a_service>. Luettu 27.2.2012.

2. Järvi, Antero; Karttunen, Jussi; Mäkilä, Tuomas; Ipatti, Jouni. 2011. SaaS-Käsikirja. Turun yliopisto, Tekes.

3. Advantages of SaaS (Software as a Service). 2010. Verkkodokumentti. CloudTweaks. <http://www.cloudtweaks.com/2010/08/advantages-of-saas-software-as-a-service/>. Luettu 28.2.2012.

4. Pilvilaskenta – Cloud Computing. 2012. Verkkodokumentti. <http://pilvilaskenta.wikispaces.com/Pilvilaskenta+-+Cloud+Computing>. Luettu 30.3.2012.

5. Schmidt, Klaus. 2006. High Availability and Disaster Recovery: Concepts, Design, Implementation. Springer.

6. Multitenantti arkkitehtuuri. 2012. Verkkodokumentti. <http://pilvilaskenta.wikispaces.com/Multitenantti-arkkitehtuuri>. Luettu 30.3.2012.

7. Gil, Paul. 2010. What Is 'SaaS' (Software as a Service). Verkkodokumentti. <http://netforbeginners.about.com/od/s/f/what_is_SaaS_software_as_a_service.htm>. Luettu 28.2.2012

8. Pan, Jiantao. 1999. Software Reliability. Verkkodokumentti. <http://www.ece.cmu.edu/~koopman/des_s99/sw_reliability/>. Luettu 29.3.2012.

9. Al-Sahhar, Rami. 2009. Software As A Service – SaaS: The Future of Flexible Software Model. Verkkodokumentti. <http://www.slideshare.net/SmartManQ8/saas-1597107>. Luettu 29.3.2012

10. Everything Should Be Made as Simple as Possible, But Not Simpler. 2012. Verkkodokumentti. Quote Investigator. <http://quoteinvestigator.com/2011/05/13/einstein-simple/>. Luettu 2.4.2012

11. Redundancy (engineering). Wikipedia tietosanakirja. <http://en.wikipedia.org/wiki/Redundancy_%28engineering%29>. Luettu 2.4.2012

12. Preston, Curtis W. 2007. Backup & Recovery: Inexpensive Backup Solutions for Open Systems. O'Reilly.

13. Hodges, Robert & Pipes, Jay. 2010. Not Your Grandpa's Replication: The New Wave of MySQL Replication and How It Helps Your Applications. Verkkodokumentti. <http://assets.en.oreilly.com/1/event/36/Not%20Your%20Grandpa%E2%80%99s%20Replication-The%20New%20Wave%20of%20MySQL%20Replication%20and%20How%20It%20Helps%20Your%20Applications%20Presentation.pdf>. Luettu 3.4.2012

14. Cron. Wikipedia tietosanakirja. <http://en.wikipedia.org/wiki/Cron>. Luettu 4.4.2012.

15. Task Scheduler. Verkkodokumentti. <http://msdn.microsoft.com/en-us/library/windows/desktop/aa383614%28v=vs.85%29.aspx>. Luettu 4.4.2012.

16. Dowling, Ben. 2009. 10 Free Server & Network Monitoring Tools that Kick Ass. Verkkodokumentti. <http://sixrevisions.com/tools/10-free-server-network-monitoring-tools-that-kick-ass/>. Luettu 3.4.2012.

17. Loker, George. 2011. Backup, Backup, Backup! Verkkodokumentti. <http://staff.drewloker.com/backup.htm>. Luettu 25.3.2012.

18. Garfinkel, Simson & Spafford, Gene. 2003. Practical UNIX & Internet Security. O'Reilly.

19. Failover. Wikipedia tietosanakirja. <http://en.wikipedia.org/wiki/Failover>. Luettu 4.4.2012.

20. Typing Tutor and Typing Test Programs. Verkkosivu. TypingMaster. <http://www.typingmaster.com/>. Luettu 29.2.2012.

21. TypingMaster Online. Verkkosivu. TypingMaster <http://www.typingmaster.com/education/>. Luettu 29.2.2012.

22. AssessTyping.com. Verkkosivu. TypingMaster. <http://www.assesstyping.com/>. Luettu 29.2.2012.

23. TypingTest.com. Verkkosivu. TypingMaster. <http://www.typingtest.com/>. Luettu 29.2.2012.

24. Justin. 2008. Data Integrity? Kommentti verkkokirjoitukseen. <http://forums.theregister.co.uk/forum/1/2008/02/07/freeformdynamics_saas_reality_check/>. Luettu 13.3.2012.

25. Multi-Master replication in MySQL 5.1. 2009. Verkkodokumentti. MySQL. <http://forums.mysql.com/read.php?26,276312,276312>. Luettu 12.3.2012.

26. A simple BASH script to help overclockers. 2010. Verkkodokumentti. <http://www.overclock.net/t/643843/a-simple-bash-script-to-help-overclockers>. Luettu 15.3.2012.

27. Viklund, Andreas. 2012. psad: Intrusion Detection and Log Analysis with iptables. Verkkodokumentti. <http://cipherdyne.org/psad/>. Luettu 30.3.2012.

28. Viklund, Andreas. 2012. fwsnort: Application Layer IDS/IPS with iptables. Verkkodokumentti. <http://cipherdyne.org/fwsnort/>. Luettu 30.3.2012

29. Windows Server 2003 can take you back in time. 2005. Verkkodokumentti. <http://blogs.msdn.com/b/oldnewthing/archive/2005/09/06/461390.aspx>. Luettu 25.3.2012.

DBBackup Bash-skripti

```
#!/bin/sh
workdir=/root/dbbackups/
dbname=$1

find ${workdir}${dbname}/${dbname}*.* -mtime +30 -exec rm {} \;

lastname=$(ls -tr ${workdir}${dbname}/${dbname}*.* | tail -1)
lastsize=$(du -b $lastname | sed 's/\([0-9]*\)\(.*\)/\1/')

newname=${workdir}${dbname}/${dbname}_$(date +%Y-%m-%d_%H.%M).sql.gz
mysqldump -u'user' -p'password' ${dbname} | gzip ->$newname

newsize=$(du -b $newname | sed 's/\([0-9]*\)\(.*\)/\1/')
if [ $newsize -le $lastsize ]; then
        echo "The database backup file for ${dbname} did not grow from yester-
        day. Check that replication and /etc/dbbackup are working." | mail -s
        "Database backup problem for ${dbname}" alerts@emailaddress.com
fi
```

SaaS-varmuuskopioiden latausskriptit Extranet-palvelimelle

TypingTest.bat

```
c:
cd \tools\winscp
winscp.com /console /script=c:\scripts\typingtest.txt 2 > errors-typingtest.log
cd c:\backups\daily\typingtest\
forfiles /p c:\backups\daily\typingtest\ /m *.* /d -90 /c "cmd /c del @file"
```

TypingTest.txt

```
option batch on
option confirm off
open user@0.0.0.0
cd /root
option transfer binary
synchronize local c:\backups\daily\typingtest\ dbbackups
close
exit
```

TMOnline.bat

```
c:
cd \tools\winscp
winscp.com /console /script=c:\scripts\tmonline.txt 2>errors-tmonline.log
cd c:\backups\daily\tmonline\
forfiles /p c:\backups\daily\tmonline\ /m *.* /d -90 /c "cmd /c del @file"
cd c:\backups\daily\assesstyping\
forfiles /p c:\backups\daily\assesstyping\ /m *.* /d -30 /c "cmd /c del @file"
cd c:\backups\daily\tmonlinelms\
forfiles /p c:\backups\daily\tmonlinelms\ /m *.* /d -30 /c "cmd /c del @file"
cd c:\backups\daily\tmonlinesingle\
forfiles /p c:\backups\daily\tmonlinesingle\ /m *.* /d -30 /c "cmd /c del @file"
```

TMOnline.txt

```
option batch on
```

```
option confirm off
open user@0.0.0.0
cd /root/dbbackups
option transfer binary
synchronize local c:\backups\daily\tmonline\ tmonline
synchronize local c:\backups\daily\assesstyping\ assesstyping
synchronize local c:\backups\daily\tmonlinesingle\ tmonlinesingle
synchronize local c:\backups\daily\tmonlinelms\ tmonlinelms
close
exit
```

Monthly Backup Batch-skripti

```
@echo off
setlocal

set source=c:\backups\daily\assesstyping
set destdir=c:\backups\monthly\assesstyping
pushd "%source%"
for /f "tokens=*" %%a in ('dir *.* /b /a-d /o:e 2^>NUL') do (set lfile=%%a)
echo copying "%source%\%lfile%" to "%destdir%"
copy /y "%source%\%lfile%" "%destdir%"

set source=c:\backups\daily\tmonline
set destdir=c:\backups\monthly\tmonline
pushd "%source%"
for /f "tokens=*" %%a in ('dir *.* /b /a-d /o:e 2^>NUL') do (set lfile=%%a)
echo copying "%source%\%lfile%" to "%destdir%"
copy /y "%source%\%lfile%" "%destdir%"

set source=c:\backups\daily\tmonlinelms
set destdir=c:\backups\monthly\tmonlinelms
pushd "%source%"
for /f "tokens=*" %%a in ('dir *.* /b /a-d /o:e 2^>NUL') do (set lfile=%%a)
echo copying "%source%\%lfile%" to "%destdir%"
copy /y "%source%\%lfile%" "%destdir%"

set source=c:\backups\daily\tmonlineng
set destdir=c:\backups\monthly\tmonlineng
pushd "%source%"
for /f "tokens=*" %%a in ('dir *.* /b /a-d /o:e 2^>NUL') do (set lfile=%%a)
echo copying "%source%\%lfile%" to "%destdir%"
copy /y "%source%\%lfile%" "%destdir%"

set source=c:\backups\daily\tmonlinesingle
```

```
set destdir=c:\backups\monthly\tmonlinesingle
pushd "%source%"
for /f "tokens=*" %%a in ('dir *.* /b /a-d /o:e 2^>NUL') do (set lfile=%%a)
echo copying "%source%\%lfile%" to "%destdir%"
copy /y "%source%\%lfile%" "%destdir%"

set source=C:\backups\tmserver2\dev-company-marketing-home
set destdir=c:\backups\tmserver2-monthly\dev-company-marketing-home
pushd "%source%"
echo copying "%source%\*normal*" to "%destdir%"
copy /y "%source%\*normal*" "%destdir%"

set source=C:\backups\tmserver2\archive
set destdir=c:\backups\tmserver2-monthly\archive
pushd "%source%"
echo copying "%source%\*normal*" to "%destdir%"
copy /y "%source%\*normal*" "%destdir%"
```

SaaS-palveluiden kuukausittaisten varmuuskopioiden latausskriptit Intranet-palvelimelle

monthly-backup.bat

```
c:
cd \tools\winscp
winscp.com /console /script=c:\tools\winscp\monthly-backup.txt 2 > errors-monthly-backup.log
```

monthly-backup.txt

```
option batch on
option confirm off
open user@0.0.0.0
cd backups\monthly
option transfer binary
synchronize local F:\Backups\tmonline_monthly\ tmonline3
synchronize local F:\Backups\tmonline_monthly\ tmonline
synchronize local F:\Backups\tmonline_monthly\ assesstyping
synchronize local F:\Backups\tmonline_monthly\ tmonlineng
synchronize local F:\Backups\tmonline_monthly\ tmonlinesingle
synchronize local F:\Backups\tmonline_monthly\ tmonlinelms
close
exit
```

Asiakasdatan varmuuskopiointiskriptit Intranet-palvelimelle

extranet.bat

```
set NOW=%date:~10,4%-%date:~4,2%-%date:~7,2%

"C:\Program Files\MySQL\MySQL Server 5.5\bin\mysqldump" --user=user --
password=password tmsalesdb > C:\backups\extranet\tmsalesdb.sql
"C:\Program Files\MySQL\MySQL Server 5.5\bin\mysqldump" --user= user --
password=password flyspray > C:\backups\extranet\flyspray.sql
"C:\Program Files\MySQL\MySQL Server 5.5\bin\mysqldump" --user= user --
password=password helpdesk > C:\backups\extranet\helpdesk.sql
"C:\Program Files\MySQL\MySQL Server 5.5\bin\mysqldump" --user= user --
password=password kbase > C:\backups\extranet\kbase.sql
"C:\Program Files\MySQL\MySQL Server 5.5\bin\mysqldump" --user= user --
password=password kb_fin > C:\backups\extranet\kb_fin.sql
"C:\Program Files\MySQL\MySQL Server 5.5\bin\mysqldump" --user= user --
password=password keygen_log > C:\backups\extranet\keygen_log.sql

"C:\Program Files\7-Zip\7z.exe" a "C:\backups\extranet\extranet-mysqldump-
%NOW%.sql.7z" "C:\backups\extranet\*.sql"

cd c:\backups\extranet\
del *.sql

forfiles /p C:\backups\extranet\ /m *.* /d -14 /c "cmd /c del @file"
```

Batch-skriptit verkkolevyjen varmuuskopiointia varten

backup-tm2.bat

```
set NOW=%date:~10,4%-%date:~4,2%-%date:~7,2%
REM SVN hotcopy
F:
cd \tmserver2_backups\temp\
rd /q /s svnbackup
md svnbackup
"C:\Program Files\VisualSVN Server\bin\svnadmin.exe" hotcopy C:\svnroot\main\
F:\Backups\tmserver2_backups\temp\svnbackup\
if not errorlevel 0 goto error

REM NTBackup
C:
C:\WINdows\system32\ntbackup.exe backup "@C:\Documents and Set-
tings\Administrator\Local Settings\Application Data\Microsoft\Windows
NT\NTBackup\data\development.bks" /n "TMSERVER2 %1 development, company,
home & marketing backup" /d "TMSERVER2 %1 development, company, home & mar-
keting backup" /v:no /r:no /rs:no /hc:off /m %1 /j "%1 development, company, home
& marketing backup" /l:f /f "F:\Backups\tmserver2_backups\dev-company-marketing-
home\tm2-dev-company-marketing-home-%1-%NOW%.bkf"
if not errorlevel 0 goto error

REM 7zip
c:\tools\7zip\7za.exe a "F:\Backups\tmserver2_backups\dev-company-marketing-
home\tm2-dev-company-marketing-home-%1-%NOW%.bkf.7z"
"F:\Backups\tmserver2_backups\dev-company-marketing-home\tm2-dev-company-
marketing-home-%1-%NOW%.bkf" -ppassword
if not errorlevel 0 goto error

REM Delete the unzipped temp file
del "F:\Backups\tmserver2_backups\dev-company-marketing-home\tm2-dev-company-
marketing-home-%1-%NOW%.bkf"
```

```
REM WinSCP
start "Dev + Company + Marketing + Home backups" C:\tools\winscp\tm2-upload.bat
dev-company-marketing-home

REM NTBackup
C:
C:\WINdows\system32\ntbackup.exe backup "@C:\Documents and Set-
tings\Administrator\Local Settings\Application Data\Microsoft\Windows
NT\NTBackup\data\archive.bks" /n "TMSERVER2 %1 archive backup" /d "TMSERVER2
%1 archive backup" /v:no /r:no /rs:no /hc:off /m %1 /j "%1 archive backup" /l:f /f
"F:\Backups\tmserver2_backups\archive\tm2-archive-%1-%NOW%.bkf"
if not errorlevel 0 goto error

REM 7zip
c:\tools\7zip\7za.exe a F:\Backups\tmserver2_backups\archive\tm2-archive-%1-
%NOW%.bkf.7z F:\Backups\tmserver2_backups\archive\tm2-archive-%1-
%NOW%.bkf -ppassword
if not errorlevel 0 goto error

REM Delete the unzipped temp file
del F:\Backups\tmserver2_backups\archive\tm2-archive-%1-%NOW%.bkf

REM WinSCP
start "Archive backups" C:\tools\winscp\tm2-upload.bat archive

REM Download Extranet MySQLDumps
c:
cd c:\tools\winscp
winscp.com /console /script=c:\tools\winscp\extranet.txt 2 > errors-extranet.log

goto end

REM In case of error
```

```
:error
cd c:\tools\sendmail\
sendmail.exe -t < error_backup.txt
goto end

:end
```

upload-tm2.bat

```
@ECHO OFF

set counter=0

:upload
c:
cd c:\tools\winscp
echo a | winscp.com /console /script=c:\tools\winscp\%1.txt 2 > errors-%1.log
if errorlevel 1 goto error

forfiles /p F:\Backups\tmserver2_backups\dev-company-marketing-home\ /m *.* /d -7
/c "cmd /c del @file"
forfiles /p F:\Backups\tmserver2_backups\archive\ /m *.* /d -7 /c "cmd /c del @file"
goto end

:error
set /a counter+=1
if %counter% GTR 24 goto sendmail
                echo Error!
                sleep 600
                echo Resuming upload...
                goto upload

:sendmail
cd c:\tools\sendmail\
sendmail.exe -t < error_sftp.txt
```

```
goto end

:end
exit
```

dev-company-marketing-home.txt

```
option batch abort
option confirm off
open user@0.0.0.0
cd backups
cd tmserver2
option transfer binary
option exclude "svnbackup"
option exclude "temp"
option include "*.bkf.7z"
synchronize remote F:\Backups\tmserver2_backups\dev-company-marketing-home\
dev-company-marketing-home
close
exit
```

archive.txt

```
option batch abort
option confirm off
open user@1.1.1.1
cd backups
cd tmserver2
option transfer binary
option exclude "svnbackup"
option exclude "temp"
option include "*.bkf.7z"
synchronize remote F:\Backups\tmserver2_backups\archive\ archive
close
exit
```

tm2-backup-cleanup.bat

```
c:
cd C:\backups\tmserver2\archive
forfiles /p C:\backups\tmserver2\archive /m *differential*.* /d -7 /c "cmd /c del @file"

cd C:\backups\tmserver2\dev-company-marketing-home
forfiles /p C:\backups\tmserver2\dev-company-marketing-home /m *differential*.* /d -7 /c "cmd /c del @file"

cd C:\backups\tmserver2\archive
forfiles /p C:\backups\tmserver2\archive /m *normal*.* /d -60 /c "cmd /c del @file"

cd C:\backups\tmserver2\dev-company-marketing-home
forfiles /p C:\backups\tmserver2\dev-company-marketing-home /m *normal*.* /d -60 /c "cmd /c del @file"
```

Raidmonitor Bash-skripti

```
#!/bin/bash

HOSTNAME=$(hostname -f)
FULLSTATUS=$(/etc/tw_cli /c0 show all)

com=`/etc/tw_cli info c0 u0 status | awk '{print $4}'`
echo $com

if [ "$com" != "OK" ]; then
        echo –e "RAID problem detected on $HOSTNAME\n\nFull controller sta-
        tus:\n$FULLSTATUS\n\nPlease resolve this ASAP." >> /tmp/body
        mail -s ">> RAID Warning on $HOSTNAME <<"
        alerts@emailaddress.com < /tmp/body
        rm -f /tmp/body
fi
```

Sensormonitor Bash-skripti

```
#!/bin/bash

ALARMADDRESS=alerts@emailaddress.com

temp1=`sensors | grep -e 'Core 0' |cut -c15,16`
temp2=`sensors | grep -e 'Core 1' |cut -c15,16`
temp3=`sensors | grep -e 'Core 2' |cut -c15,16`
temp4=`sensors | grep -e 'Core 3' |cut -c15,16`
fan1=`sensors | grep -e 'fan1' | awk 'NR==2' | cut -c13,14,15,16`
fan5=`sensors | grep -e 'fan5' | cut -c13,14,15,16`

if [ "$temp1" -ge "80" ]; then
                msg="CPU, Core 0 temperature alert: $temp1 degrees!"
elif [ "$temp2" -ge "80" ]; then
                msg="CPU, Core 1 temperature alert: $temp2 degrees!"
elif [ "$temp3" -ge "80" ]; then
                msg="CPU, Core 2 temperature alert: $temp3 degrees!"
elif [ "$temp4" -ge "80" ]; then
                msg="CPU, Core 3 temperature alert: $temp4 degrees!"
elif [ "$fan1" -eq "0" ]; then
                msg="Fan1 alert: Fan1 has stopped!"
elif [ "$fan5" -eq "0" ]; then
                msg="Fan5 alert: Fan5 has stopped!"
fi

if [ "$msg" != "" ]; then
                subject="$HOSTNAME $msg"
                message="Host $HOSTNAME reports a sensor alert: $msg"
                echo "$message" | mail -s "$subject" "$ALARMADDRESS"
fi
```

Verkkolevyn palautusohjeet

Backup palautus

- Ota sFTP-yhteys Extranet-serveriin (esim. WinSCP:llä).
- Siirrä halutut backupit kansiosta C:\backups\tmserver2\ TMServer2:lle tai itsellesi. Siirrä joko 1 full backup ja tarvittavat differentiaalit, tai pelkät differentiaalit, riippuen minkä päivän backupit haluat. Useaa full backupia ei tarvitse siirtää.
- Pura backupit 7zip-työkalulla. Command Prompt: 7za x <tiedosto>.7z ja syötä salasana, tai sama graafisen työkalun avulla.
- Avaa NTBackup-työkalu. Mikäli käytössä on Windows 7 tai Windows 2008, tulee tämä hakea erikseen kansiosta S:\tmserver2\ntbackup\ (ntbackup.exe, ntmsapi.dll ja vssapi.dll).
- Restore files and settings
- Browse, valitse purettu bkf-tiedosto
- Items to restore: ruksaa haluamasi kansio(t)
- Advanced
- Restore files to: Alternate location, valitse kansio johon haluat palauttaa tiedostot
- Leave existing files
- Preserve existing volume mount points ruksi pois
- Windows 7:ssa NTBackup-työkalu antaa error-raportin, siitä ei tule välittää.
- Nyt tiedostot on palautettu.

Täysi backup palautus

- Ota Intranetistä sFTP-yhteys Extranet-serveriin (esim. WinSCP:llä).
- Siirrä halutut backupit kansiosta C:\backups\tmserver2\ TMServer2:lle. Siirrä uusin full backup ja kaikki tämän jälkeiset differentiaalit.
- Pura backupit 7zip-työkalulla. Command Prompt: 7za x <tiedosto>.7z ja syötä salasana, tai sama graafisen työkalun avulla.
- Avaa NTBackup-työkalu
- Restore files and settings
- Browse, valitse purettu bkf-tiedosto

- Items to restore: ruksaa haluamasi kansio(t)
- Advanced
- Restore files to: Original location
- Restore security settings ja Preserve existing volume mount points ruksattuna
- Nyt tiedostot on palautettu.